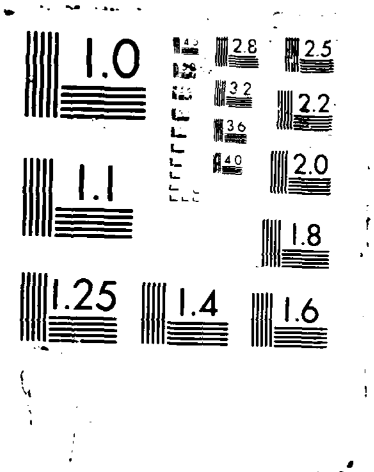
AD-A190 602 A SIMULATION MODEL OF THE ASD (AERONAUTICAL SYSTEMS 1/2
DIVISION) CENTRAL DAT..(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. R D SMEDLEY
UNCLASSIFIED DEC 87 AFIT/GE/ENG/87D-61 F/G 12/7 NL


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

A SIMULATION MODEL OF THE ASD

CENTRAL DATACOMM SYSTEM (CDS)

THESIS

Richard D. Smedley, Jr.
Captain, USAF

AFIT/GE/ENG/87D-61

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/87D-61

A SIMULATION MODEL OF THE ASD CENTRAL DATACOMM SYSTEM (CDS)

THESIS

Richard D. Smedley, Jr.
Captain, USAF

AFIT/GE/ENG/87D-61

Approved for public release; distribution unlimited

AFIT/GE/ENG/87D-61

A SIMULATION MODEL OF THE ASD CENTRAL DATACOMM SYSTEM (CDS)

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

Richard D. Smedley, Jr., B.S.
Captain, USAF

December 1987

Approved for public release; distribution unlimited

Acknowledgements

The completion of this thesis effort would not have been possible without the help of others. I am deeply indebted to my faculty advisor, Lt Col Albert B. Garcia, for his guidance, assistance, and patience during this effort. I also wish to thank Capt Wade Shaw, member of my thesis committee, for his professional guidance. I am also appreciative to the many ASD Information Systems and Technology Center members for their help and patience during this research, but I am especially grateful to Dr. Robert Jurick for his cooperation. I would be remiss if I did not mention the help I received from the CDS designers at Control Data Corporation, especially Karl Vogel and Randy Barker. Finally, I will forever be indebted to my wife Carol and my children Aimee and Sean for their understanding and concern on the many occassions when they had to take a back seat to this thesis.

Richard D. Smedley, Jr.

Table of Contents

List of Figures

List of Tables

AFIT/GE/ENG/87D-61

Abstract

The purpose of this effort was to develop a simulation model of the CDS, a complex data communications network, using Simulation Language for Alternative Modeling (SLAM) II. The study had two objectives:

(1) Determine if the current design can support the projected Phase I workload. This determination is made after looking at the local packet delay, the availability of input and output lines, and the utilization of the various components.

(2) Determine the effect of increased DDN traffic on the CDS. The specific effects studied are the packet delay within the DDN gateways, the local packet delay, and the total CDS delay on DDN packets.

Since the CDS was not operational during the model development, there were no CDS statistics available to develop the workload; therefore, the input data driving the model was derived from the workload on the current computer systems. This analysis included the basic steps of collecting the data, forming a histogram, making a distributional assumption, and using the chi-square goodness-of-fit test to accept or reject the assumptions.

The CDS simulation model demonstrated the ability of the CDS to support the Phase I workload. Additionally, the model verified that SLAM II can be used to model complex communications networks.

A SIMULATION MODEL OF THE ASD CENTRAL DATACOMM SYSTEM (CDS)

I. Introduction

Background

The Aeronautical Systems Division's (ASD) Information Systems and Technology Center (ISTC) provides general purpose computer support to all ASD organizations, the Air Force Aerospace Medical Research Laboratory, the Air Force Institute of Technology, the Air Force Human Resources Laboratory, and the Air Force Systems Command Procurement Management Offices located throughout the United States. Over the years, support of these diverse communities has led to the installation of many different types of computer systems. The differing computer systems led to a myriad of connections between the users and the ISTC systems. As the user community expanded, so did the complexity of the data communications equipment. This complexity in turn brought about the need for more people and equipment to support the systems. The computer systems managers at the ISTC realized that as the mission continued to expand, so would the reliance on the computer systems; therefore, the communications interface to the ISTC systems had to be simplified. This need led to the solicitation of proposals for the installation of a Central Datacomm System (CDS).

The CDS is the central interface between the user stations and the computer systems at the ISTC. Its primary goals are as follows (1:12):

1. Reduce the complexity of the ISTC's data communications equipment.

2. Reduce the resources (people, equipment, etc.) required to support datacomm on each of the computers within the ISTC.

3. Reduce the number of connections between the user stations and the ISTC computer systems.

4. Act as the single point of entry to the ISTC resources; thus, providing a single point of control for migration and implementation of new datacomm technology, networking standards, and security access procedures and standards.

5. Serve as the primary interface between the user stations and the Defense Data Network (DDN).

The CDS is the heart of the data communications supporting the ISTC computer resource. If the CDS is saturated by a given workload, service to all the systems is seriously degraded; therefore, the ISTC system managers asked the Electrical Engineering Department at the Air Force Institute of Technology (AFIT) to evaluate the proposed CDS network.

There are two compelling reasons a CDS simulation model is needed. First, much of the design was done on an ad-hoc basis. Intuition and past experience were the key methods used to develop the CDS design (10:2). Thus, the simulation model helps to show the proposed design can support the specified workload.

Second, if the CDS is going to serve as ASD's primary interface to the DDN world, it is imperative the CDS be able

to provide acceptable service. Specifically, the CDS design is based on four 56 Kbps data links between the CDS and the DDN. The simulation model tests the practicality of using the CDS as the DDN gateway based on this link speed. Additionally, the CDS committee wants to test the gateway concept when each DDN link speed is increased to 1.544 megabits per second (T1) (6,16).

Central Datacomm System (CDS) Overview

The CDS is a modular system "designed around industry standard components" (8:Sec C,3) to insure the previously mentioned goals are met. The modularity concept comes from the use of multiple super-mini computers as the building blocks of the CDS. This type of approach allows phased growth "with little modification to the basic CDS system" (8:Sec C,3). Thus, as the user community and user needs expand, the additional requirements can be met by adding more building blocks. This feature is important because growth is going to be an integral part of the future of the CDS. Recognizing this fact, the CDS committee divided the installation of the CDS into three distinct phases.

Phase I covers the first 18 months of the contract and Table 1 specifies the input and output connections for Phase I. Phase II covers the next 18 months and Phase III covers the final 24 months of the contract. The Phase II and Phase III requirements expand or modify the Phase I requirements, but the exact details for these phases are still being finalized. Since the only detailed information

Table 1. The Supported Phase I CDS Configuration (Source 8)

User Connection to CDS			Max Number of Simultaneous Users		CDS Connection to SRF Resources			Max Number of Simultaneous Users
Protocol	Baud Rate	Source			Protocol	Baud Rate	Resource	
ASCII/TTY	1200*	Dial-Up	52		ASCII/TTY	Auto-2400	CDC	4
ASCII/TTY	1200*	Multiplexed	45		ASCII/TTY	Auto-9600	CDC	4
ASCII/TTY	2400*	Dial-Up	5		X.25	Auto-9600	CDC	108
ASCII/TTY	2400*	Multiplexed	90		HASP	19200	CDC	10
ASCII/TTY	Auto-9600	Dial-Up	0		DDN-Telnet	Ethernet	CDC	82
ASCII/TTY	Auto-9600	Multiplexed	9		DDN-FTP	Ethernet	CDC	6
ASCII/TTY	Auto-19200	Multiplexed	2					
HASP	4800	Dial-Up	2		ASCII/TTY	Auto-2400	IBM	12
HASP	9600	Multiplexed	4	C	ASCII/TTY	Auto-9600	IBM	12
HASP	19200	Multiplexed	5		X.25	Auto-9600	IBM	52
DDN-Telnet	1200	DDN	2	D				
DDN-Telnet	9600+,RS232	WPAFB LAN	9		X.25	56000	IBM	36
DDN-Telnet	T1	WPAFB LAN	11					
DDN-Telnet	Ethernet	WPAFB LAN	11	S				
DDN-FTP	1200	DDN	2					
DDN-FTP	Auto-9600	Multiplexed	2					
DDN-FTP	56000	DDN	5		HASP	19200	IBM	1
DDN-FTP	T1	DCTN	2		DDN-Telnet	1200	IBM	3
DDN-FTP	Ethernet	WPAFB LAN	5		DDN-FTP	56000	IBM	3
DDN-SMTP	1200	DDN	2		DDN-FTP	Arbitrary	IBM	3
DDN-SMTP	9600+,RS232	WPAFB LAN	9		ASCII/TTY	19200	Modcomp	12
DDN-SMTP	T1	WPAFB LAN	11		DDN-Telnet	Ethernet	VLCC	12
DDN-SMTP	Ethernet	WPAFB LAN	11		DDN-FTP	Ethernet	VLCC	1
DDN-SMTP	Auto-2400	Dial-Up	2		DDN-Telnet	Ethernet	S&E VAX	18
DDN-SMTP	Auto-9600	Dial-Up	2		DDN-FTP	Ethernet	S&E VAX	3
DDN-SMTP	Auto-9600	Multiplexed	2		DDN-Telnet	Ethernet	AMS	36
X.25	1200	DDN	2		DDN-FTP	Ethernet	AMS	14
X.25	Auto-4800	Dial-Up	2		DDN-Telnet	Ethernet	JLCF	29
X.25	Auto-9600	Dial-Up	2		DDN-FTP	Ethernet	JLCF	4
X.25	9600	Multiplexed	1		DDN-Telnet	Ethernet	AWC	12
X.25	19200	Multiplexed	1		DDN-FTP	56000	AWC	3
X.25	T1	DCTN	1		DDN-Telnet	Ethernet	INFOCEN	58
Arbitrary	9600	Multiplexed	1		DDN-FTP	Ethernet	INFOCEN	7
Arbitrary	T1	DCTN	1		MASSNET	Hyperchannel	CFS	4
3270 Bisync	9600	Dial-up (20)	18					
SNA SDLC	9600	Dial-up (5)	5		ASCII/TTY	Auto-9600	Sews Supp	3
					DDN-FTP	56000	Sews Supp	1
		Total	336				Total	553

available is for Phase I, the remainder of the thesis addresses Phase I of the CDS installation.

Figure 1 shows the interrelationship of the various building blocks of the CDS. The two main components of the CDS are the User Access Processor (UAP) and the Network Access Processor (NAP). The six UAPs are Tolerant Systems super-mini computers, and they provide most of the user interfaces to the CDS. Additionally, the UAPs provide X.25 and TTY connections from the CDS to the applications processors. The input/output function for these connections is handled by the Communications Interface Processor (CIP).

Each CIP can support up to 12 asynchronous TTY lines with data speeds of 19.2 Kbps or less, or 10 asynchronous TTY lines and 2 synchronous lines with data speeds of 56 Kbps. The Phase I configuration includes 5 primary CIPs per each UAP. For redundancy purposes, each CIP is dual homed to more than one UAP and the UAP pairs share a backup CIP. This redundant configuration comes into play when a component fails. This thesis effort models the fully operational configuration.

Whereas the UAPs provide the user access to the CDS, the two NAPs (Pyramid super-mini computers) provide the internetworking between the various applications processors. Additionally, the NAPs serve as the gateways to the Defense Data Network (DDN). Ethernet cable plants provide the connection between the NAPs and the other CDS components.

The CDS design includes eight separate ethernet cable plants. These cable plants are divided into two categories:

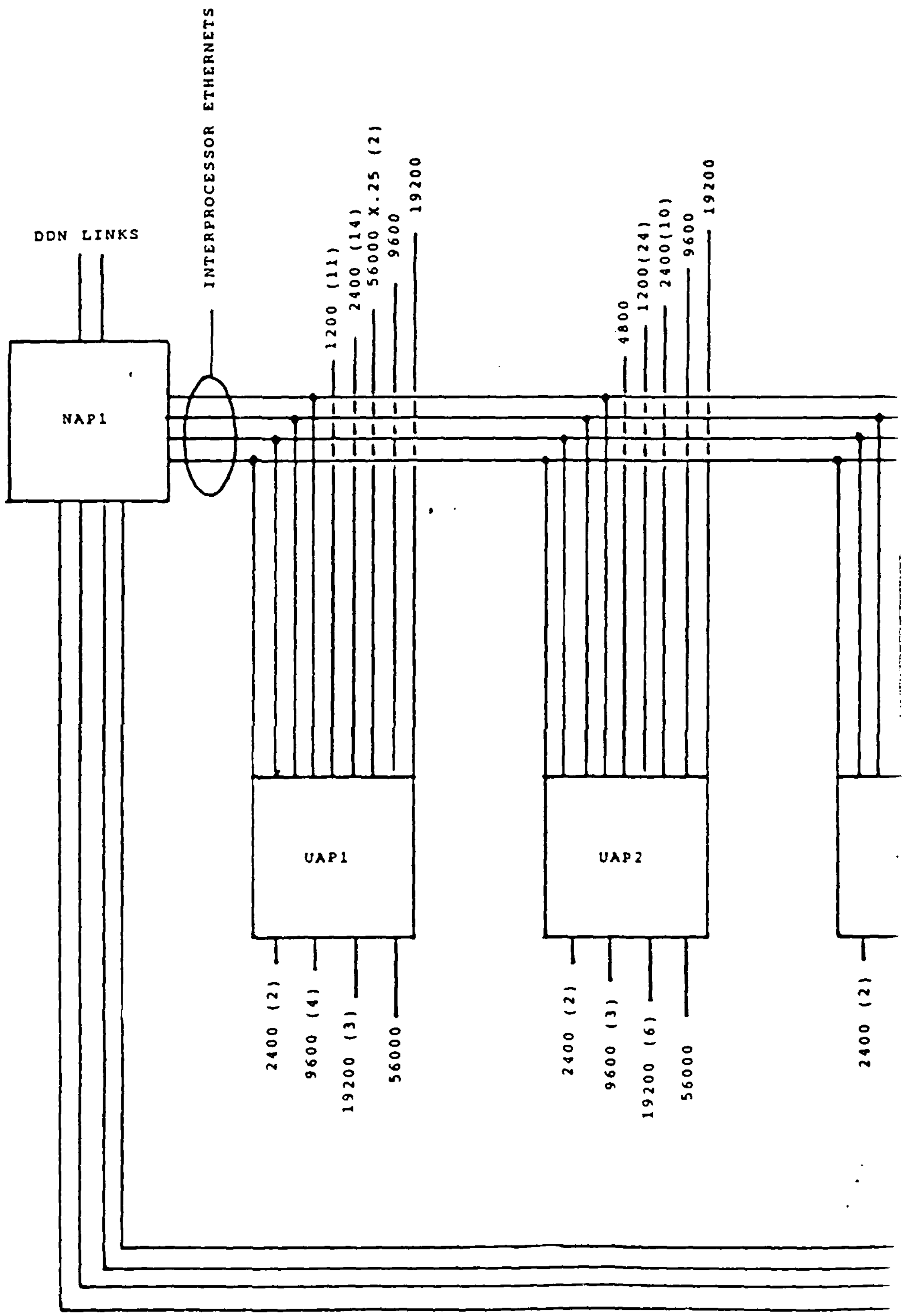
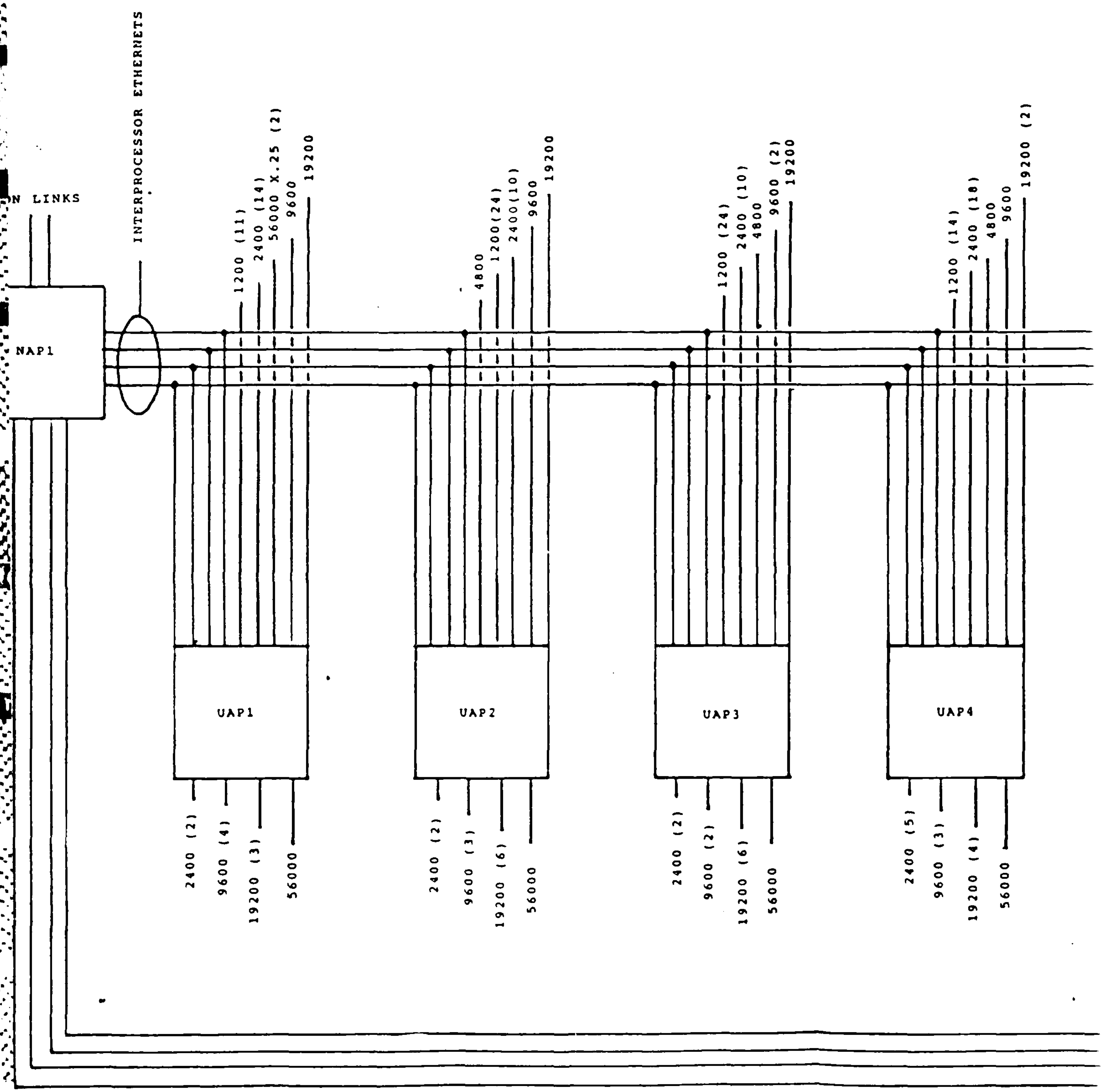
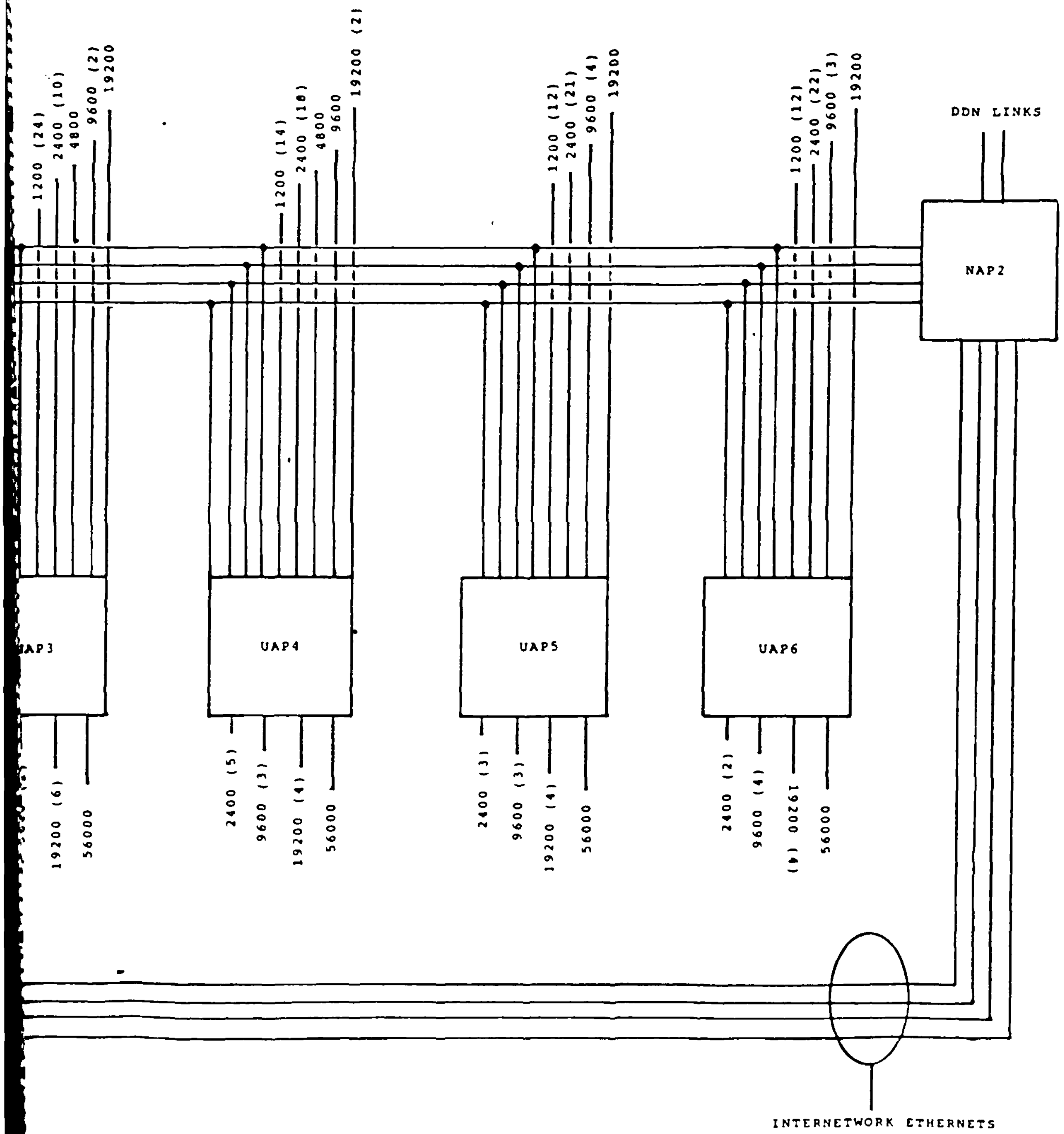


Figure 1. Block Diagram of the Major CDS Components (Source 8)



1. Block Diagram of the Major CDS Components
8)



1200 (24)
2400 (10)
4800
9600 (2)
19200
1200 (14)
2400 (18)
4800
9600
19200 (2)
1200 (12)
2400 (21)
9600 (4)
19200
1200 (12)
2400 (22)
9600 (3)
19200
DDN LINKS
NAP2
AP3
UAP4
UAP5
UAP6
19200 (6)
56000
2400 (5)
9600 (3)
19200 (4)
56000
2400 (3)
9600 (3)
19200 (4)
56000
2400 (2)
9600 (4)
19200 (4)
56000
INTERNETWORK ETHERNETS

internetwork cable plants and interprocessor cable plants. The four interprocessor ethernets connect the UAPs to each other and also connect the UAPs to the NAPs. The four internetwork ethernets connect the following systems to the NAPs: CDC Cyber computers, National Advanced Systems (NAS) computers, Modcomp computers, Cray computer (VLCC), Scientific and Engineering computer, Automated Management System (AMS) computers, Jovial Language Control Facility computer, ADA Workcenter computer, Information Central (INFOCEN) computers, Scientific and Engineering Workstation computers, and the Central File System (CFS).

Problem Statement

Develop a simulation model of the CDS to insure the current design can support the Phase I workload requirements. This determination is made after looking at the local packet delay, the availability of the input and output lines, and the utilization of the various components. Additionally, determine the effects of increased DDN traffic on the CDS. The specific effects studied are the packet delay within the DDN gateways, the local packet delay, and the total CDS delay on DDN packets.

Scope of the Research Effort

The thesis concentrates on developing a simulation model. According to Sauer and MacNair, "there are two general approaches to solutions of models of communications systems, analytic modeling, and simulation" (20:4). For a complex system like the CDS to be tractable, "analytical

models must be fairly simple in that either i) a small part of the modeled system is considered or ii) few system details are considered" (21:204). Sauer goes on to argue because of the high degree of abstraction used with most analytical models, "it is questionable whether the model has sufficient accuracy for making choices between competing designs (20:5). Therefore, the research strictly develops a simulation model of the CDS.

An important issue when developing a simulation model is what level of detail to incorporate into the model. If an extremely fine level of detail is needed, the model is developed at the micro level. This level includes all the details of the communication protocols. On the other hand if the modeler is more interested in the aggregate effect, the model is developed at the macro level (7:41-42). Since the CDS is a large and complex communications network, the model concentrates on the macro level of detail.

A macro model entails development at the host-to-host level, not down to the device level (terminals, printers, plotters, etc.). The estimated effects of the different devices are incorporated into the arrival and departure rates of the various hosts. Additionally, these rates are estimated on a per login basis. Thus, entities generated by the simulation model are logins. The simulation concentrates on the CDS components, with the various hosts attached to the CDS treated strictly as sources and sinks with respect to the model.

Assumptions

As with any simulation model, some assumptions have to be made to make the model workable. The exact assumptions used to develop the input parameters and the simulation model are listed in Chapter Two and Chapter Three respectively. Each chapter contains a summary of the assumptions made for the respective portion of the study.

Approach

The thesis effort is divided into three stages: determine the CDS data flow, develop a model based on this data flow, and exercise the model to insure the CDS can handle the Phase I workload requirements and function as the primary ASD gateway to the DDN. Figure 2 shows the general approach for completing the simulation study.

To develop the CDS data flow, it is important to understand the interrelationship of the CDS components and the ISTC computer resources. While the CDS specification and the contractor's (Control Data Corporation) response addressed each component separately, they did not initially include a top-level picture of the entire CDS network. Therefore, a graphical model is developed during this stage. This diagram shows how all components of the system are connected. Figure 1 is a scaled-down version of this diagram.

Once the physical connections are understood, it is necessary to determine the workload of each host to determine the data flow within the CDS. Any model, whether analytical

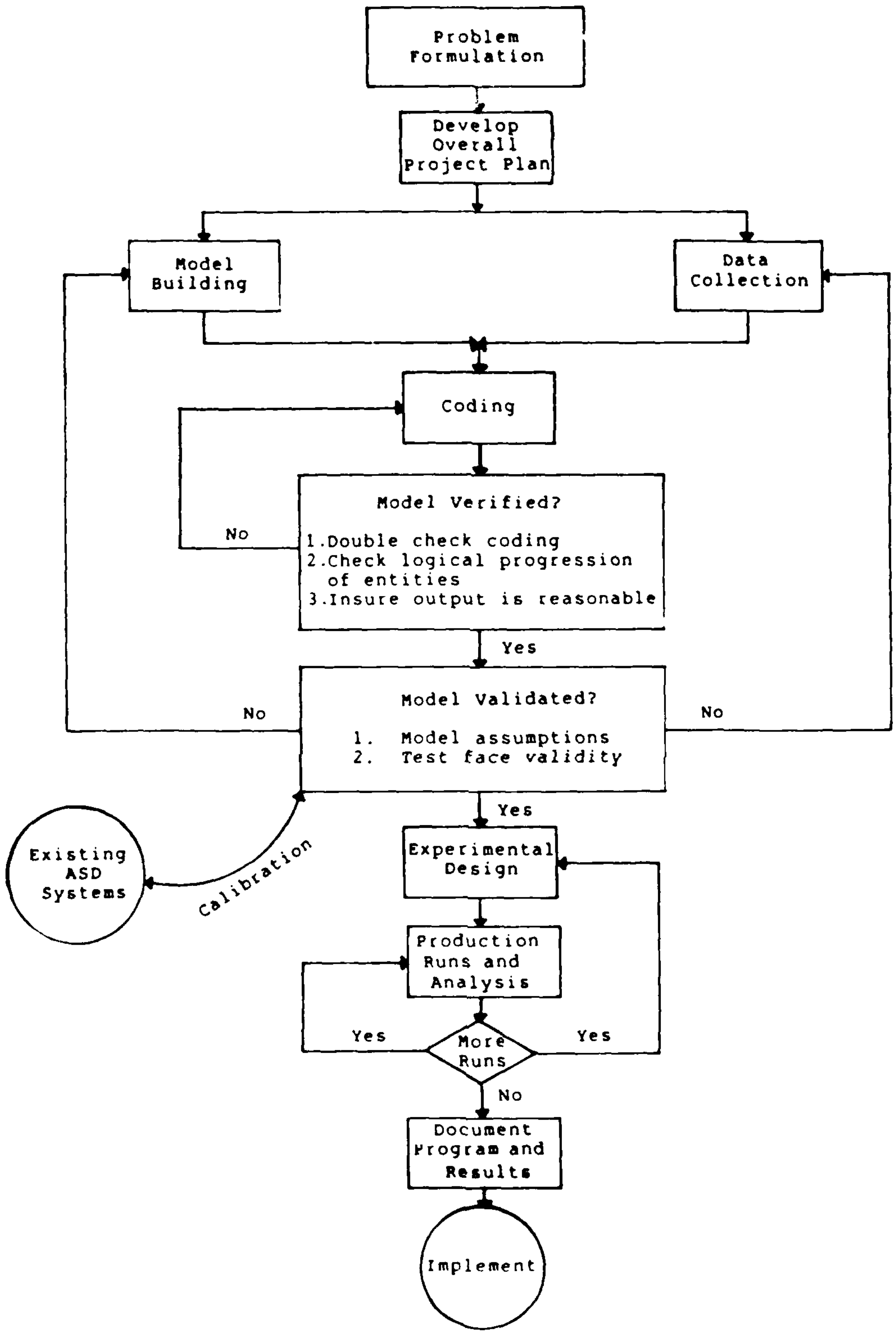


Figure 2. Steps in the CDS Simulation Study (Source 4:12)

or simulation, is built upon statistical parameters; however, collecting the necessary data to derive the statistical parameters may be difficult or even impossible. The ISTC computer network supports more than 13 host computers, and determination of exact data flow statistics is sometimes very difficult. This portion of the first stage is approached from three different angles. First, all available workload statistics from the existing systems are reviewed. Where possible, these statistics are used to determine arrival and departure rates for each system, the probability distribution function of the interarrival rates, the amount of the data transferred, the source and destination host of the data, and type of job (interactive, electronic mail, file transfers). The key to this analysis is the availability of the data. In some cases only a small portion of the data is available or it is too difficult to get.

When these figures are not available, an alternate approach is taken. The type of communications protocols being used, the line speed, and the average number of concurrent users on a given system are used to derive the necessary model parameters. Appendix D of the CDS specifications includes this information.

The above techniques handle the cases where some information is available, but even some of the tables in Appendix D of specifications are incomplete. Additionally, since the CDS is a new system, service times for the various components are not known. When data is unavailable, the study relies on the past experience of ISTC system managers

and the CDS designers to develop the necessary statistical parameters. Once all the information is analyzed, these statistics plus a knowledge of the interrelationship between the various components is used to design the simulation model.

The simulation language used in this research is Simulation Language for Alternative Modeling (SLAM) II. SLAM allows the modeler to build a network using any combination of network, discrete-event, or continuous views. The network view is used to build an extended queueing network to represent the CDS (20:33).

The next step in building the model is to code the model. Once the coding of the model is completed, the iterative process of verification and validation begins. Figure 2 shows the process flow for verification and validation. Verification is nothing more than insuring "the conceptual model is reflected accurately in the computer code" (4:379). Basically, three steps are used to verify the operational model. First, the code is always double checked to guarantee it is accurate. Second, the trace feature of SLAM is used to check the logical progression of entities. Finally, the output of the model is checked to insure the results are reasonable.

The validation process insures the model is representative of the behavior of the actual system. Since the CDS is not operational, the validation process consists of two steps. First, where possible, any model assumptions are validated. For instance, it is assumed the logins to the

CDS have exponentially distributed interarrival times. This assumption is validated by using the chi-square goodness-of-fit test on all available data. This portion of the research is discussed at greater length in Chapter Two. Second, the model is checked to insure it has high face validity. Sensitivity analysis is used to test the face validity of the model. Sensitivity analysis involves varying some of the input parameters and observing the reaction of the model to these fluctuations. Since the CDS model has many input variables, only certain critical parameters are varied by ±10%. Although verification and validation are considered distinct steps in the simulation flow diagram, they are conducted at the same time (4:383-386).

Once calibration of the model is completed, this model then becomes the baseline. The baseline model is then used to answer the questions highlighted in the problem statement. First, the baseline model is run to determine if the existing CDS design can meet the workload requirements. This determination will be made after looking at three important output statistics: the utilization of the various components, the time delay incurred by packets traversing the network, and the availability of input and output links.

Second, the model is used to determine the feasibility of using the CDS as the ASD gateway to the DDN world. The local traffic is held constant and the DDN traffic is increased. The range of interest is from 0% to 200% increase in the DDN traffic. The statistics of interest are the delay introduced by the DDN gateways and the packet delay.

Additionally, the analysis is conducted at DDN data rates of 56 Kbps and 1.544 Mbps. The aforementioned range and data rate restrictions are established by the managers at ASD (6, 16).

Order of Presentation

Chapter Two contains a discussion of how the various model parameters are developed. It includes a summary of all the assumptions used to formulate the parameters.

Chapter Three addresses the SLAM model. It explains the various modules within the model and relates these modules to actual CDS design. The chapter includes a summary of the assumptions used to simplify the model development.

Chapter Four contains an analysis of various runs of the simulation model. It includes a discussion of the different types of simulation, transient versus steady-state, how to establish the length of a single run, and how to determine the number of required runs.

Finally, Chapter Five summarizes the important insights gained about the CDS design. Some suggested follow-on efforts are listed in this chapter.

II. Workload Characterization

Background

As stated in Chapter 1, an integral part of any simulation model is the input data used to drive the model. The model may be valid but if the input data is unreliable, the validity of the output or recommendations made based on the output is highly suspect. However, determining how to represent the input data is not an easy task. Unfortunately, "there are few situations where the actions of the entities within the system under study can be completely predicted in advance" (4:122). Although most input models are not deterministic in nature, there may be some statistical distribution that describes the input parameters of interest.

Carson and Banks describe four steps in developing the input data: "collecting raw data, identifying the underlying statistical distribution, estimating the parameters, and testing for goodness of fit" (4:368). This method of attack is based on the premise the environment to be simulated can be observed. The CDS was not operational when the model was developed; therefore, the input data is derived from the workload on the current systems and any workload projections developed jointly by Control Data Corporation (CDC) and the CDS managers. Before proceeding with this analysis, it is important to define the parameters needed for the CDS simulation model.

Conceptual Structure of the CDS

A simulation model must capture the essentials of the real system, but not necessarily with a one-to-one mapping between the real system and the model (4:9). For this reason, it is important to define the features of the CDS that characterize the system. Some of these important features in turn become the parameters used to drive the model.

Ferrari divides workload characterizations into two groups: the basic workload of the components and the aggregate workload (11:42). Since the CDS model is developed at a macro level, the primary feature of interest is the interactive session or login. With respect to this feature, it is important to know the number of logins per hour to each system and the statistical distribution of the time between logins. Two other aspects of a login session help to characterize the system. First, what is the length of a login session and how is this session length distributed? Second, how much data is transferred during the average login? This traffic intensity includes electronic mail and file transfers as well as the normal data transfer taking place during an average interactive session. The next three sections describe how these parameters are derived.

Statistical Model for Logins

The statistical model for logins requires two key ingredients: the number of logins per hour and the login interarrival time distribution. The login statistics from a

Codex data switch are used to determine the interarrival time distribution. Normally, these same statistics would also be used to develop the number of logins per hour, but the Codex is not the only avenue for accessing the various computer systems. Additionally, the Codex does not provide connections to all the ASD computer systems. Therefore, the following combination of sources is used to derive the number of logins per hour: any statistics on the current systems, discussions with the CDS committee members, and Table 1.

Determining the interarrival time distribution proved to be difficult because of a lack of readily available information. None of the supported systems at ASD's computer center keep a running account of the time of the logins, but the Codex data switch has the capability to collect this information. A one-week analysis of the logins through the Codex revealed that more than 60% of the logins went to the CDC and NAS systems. The remaining logins were scattered among six other systems. Because of this tendency, the CDC and NAS statistics are used to derive the login interarrival time distribution.

When trying to identify the underlying distribution, it is useful to use a histogram in making this determination (4:335). In order to build the NAS and CDC login histograms, the data is divided into two-minute intervals. Table 2 summarizes the data used in this analysis.

Table 2. Login Interarrival Data

Time Between Logins (Minutes)	Logins Per Interval CDC	NAS
0-2	97	137
2-4	84	98
4-6	57	52
6-8	34	36
8-10	21	22
10-12	13	13
12-14	13	9
14-16	6	6
16-18	5	2
18-20	2	2
20-22	1	0
22-24	0	0
24-26	2	0
26-28	0	0
28-30	0	0
30-32	1	1
32-34	0	0
34-36	1	0
36-38	1	0
38-40	0	1
Total	338	379

Figures 3 and 4 show the histograms for the CDC and NAS systems. Based on the shape of these histograms it is assumed both distributions are exponential. Now it is necessary to test these hypotheses for goodness of fit relative to a theoretical exponential distribution. In order to conduct the goodness-of-fit tests, an estimate of the parameters must be made. For an exponential distribution the parameter of interest is the mean, λ.

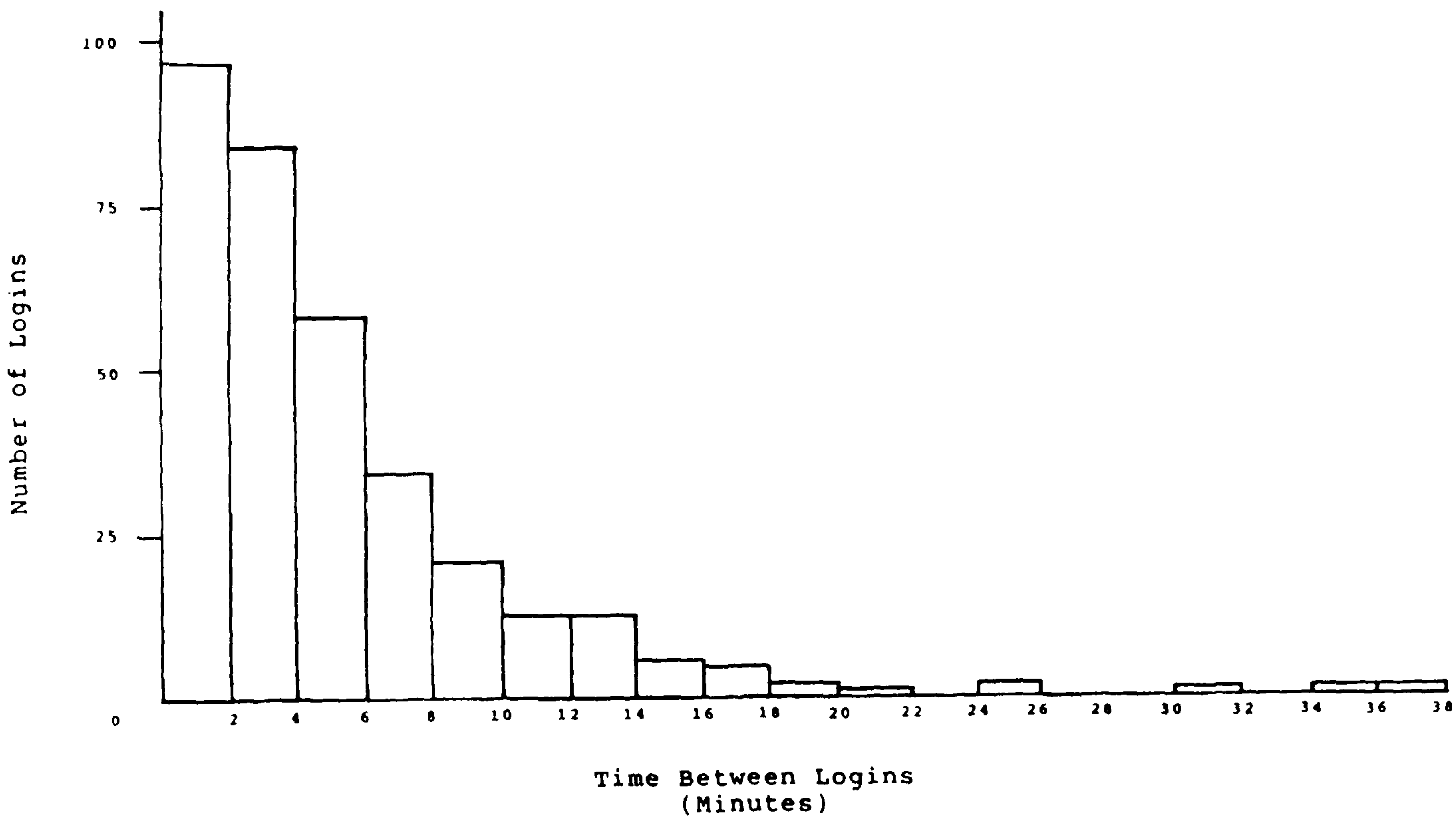


Figure 3. Histogram of the CDC Logins

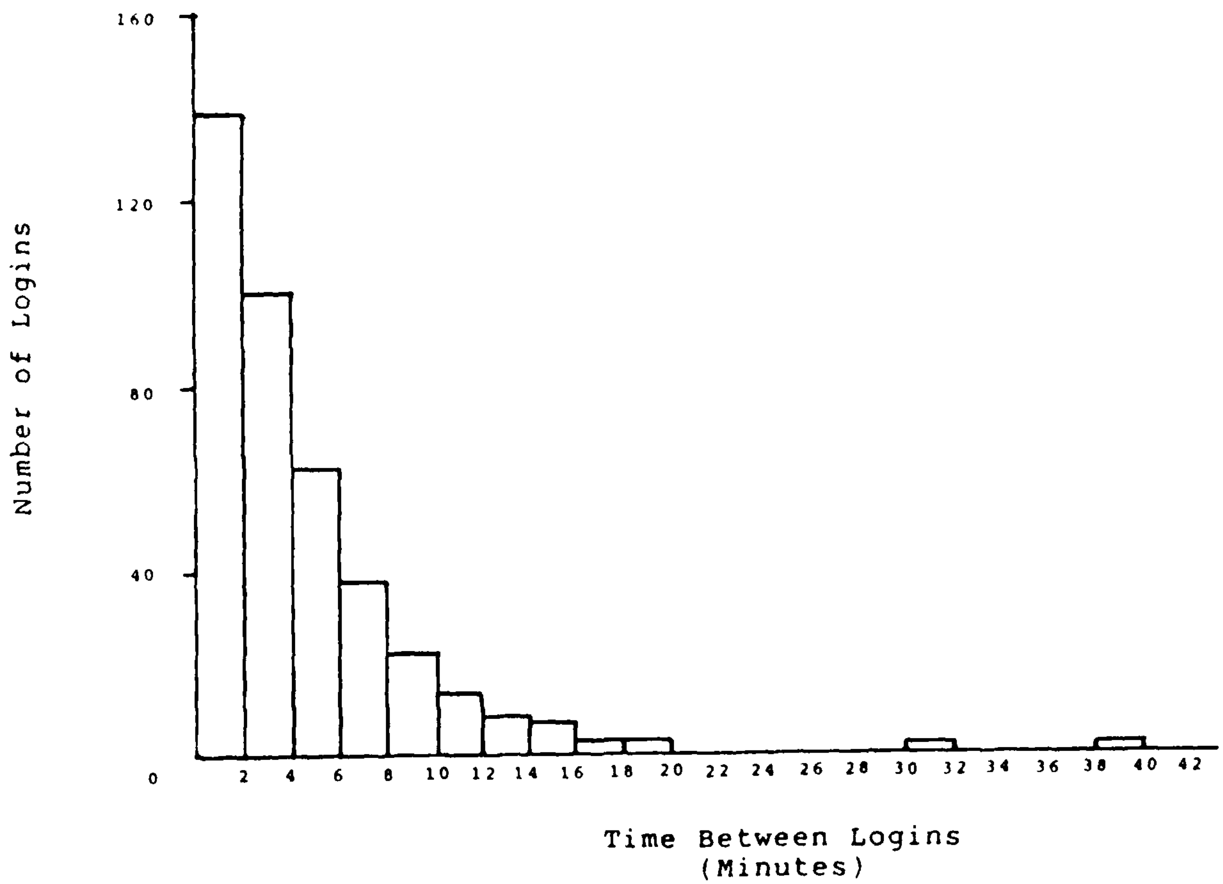


Figure 4. Histogram of the NAS Logins

A random variable X is exponentially distributed with parameter $\lambda > 0$, if its probability density function (pdf) is defined by

$$f(x) = \begin{cases} \lambda \exp[-\lambda x] , & x \geq 0 \\ 0 , & \text{elsewhere} \end{cases} \tag{1}$$

Thus, to determine the parameter λ, the sample mean is used. If the sample size is n, and the observations of these samples are X_1, X_2, . . . X_n, the sample mean ($\bar{X}$) is given by

$$\bar{X} = \sum_{i=1}^{n} X_i/n \tag{2}$$

This sample mean is used to determine the estimated parameter λ for an exponential distribution as follows:

$$\lambda = 1/\bar{X} \tag{3}$$

Applying Eqs 2 and 3 to the raw data, $\lambda = 12.588$ logins/hour for the CDC and $\lambda = 15.816$ logins/hour for the NAS. Based on these estimates, the exponential assumption can be tested.

One commonly used method for testing a distributional assumption is the chi-square goodness-of-fit test (4:290). The chi-square test is usually used when there are a large number of observations ($n > 50$). In this case the number of observations is 338 for the CDC system and 379 for the NAS

system; therefore, the chi-square test is valid for this analysis. The first step in the chi-square test is to divide the observations into k intervals. Since the histogram is divided into two-minute intervals, this arbitrary division is also used for the chi-square test, with one exception. Carson and Banks suggest an interval have a minimum of 3 to 5 observations (4:350). Therefore, if a given interval contains less than 5 observations, it is combined with adjacent intervals until the minimum requirement is satisfied. Once the intervals are established, the theoretically expected number of observations per interval is determined. Let E_i be the expected number of observations in the ith interval, then $E_i = np_i$, where p_i is the theoretical probability of the ith interval. Since the exponential distribution is continuous, p_i is computed as follows:

$$p_i = \int_{a_{i-1}}^{a_i} \lambda \exp[-\lambda x]\, dx \tag{4}$$

The test statistic for the chi-square test is given by

$$X_0^2 = \sum_{i=1}^{k} (O_i - E_i)^2 / E_i \tag{5}$$

where O_i is the number of actual observations for the ith interval. Thus, it can be shown the CDC logins interarrival time follows an exponential distribution if $X_0^2 < X^2_{\alpha,k-s-1}$, where s is the number of parameters and k is the number of

intervals for the hypothesized distribution. Since λ is the only estimated parameter for the exponential distribution, $s = 1$. Now α, the level of significance for the test, is set by the decision maker. In this case $\alpha = .05$ (4:350-351).

Applying all the above principles to the CDC raw data, the following hypotheses are formed:

H_0 : the CDC interarrival times are exponentially distributed

H_1 : the CDC interarrival times are not exponentially distributed

The chi-square test results are summarized in Table 3.

Table 3. CDC Chi-Square Goodness-Of-Fit Test Results

Interval (Minutes)	Observed Frequency, O_i	Expected Frequency, E_i	$(O_i - E_i)^2/E_i$
[0, 1.99)	97	115.83	3.060
[2, 3.99)	84	76.14	.810
[4, 5.99)	57	50.04	.970
[6, 7.99)	34	32.89	.040
[8, 9.99)	21	21.62	.020
[10, 11.99)	13	14.21	.100
[12, 13.99)	13	9.34	1.430
[14, 15.99)	6	6.14	.003
[16, 17.99)	5	4.04	.230
[18, ∞)	8	7.75	.008
Total	338	338.00	$\chi_0^2 = 6.671$

The degrees of freedom (k-s-1) for the CDC data is $10-1-1 = 8$. At $\alpha = .05$, $\chi^2_{.05,8}$ from the chi-square tables is 15.5. Since $\chi_0^2 = 6.671$, $\chi_0^2 < \chi^2_{.05,8}$. Thus, hypothesis H_0 is accepted.

The chi-square test is also applied to the following NAS hypotheses:

H_o : the NAS interarrival times are exponentially distributed

H_1 : the NAS interarrival times are not exponentially distributed

Table 4 summarizes the NAS chi-square test results.

Table 4. NAS Chi-Square Goodness-Of-Fit Test Results

Interval (Minutes)	Observed Frequency, O_i	Expected Frequency, E_i	$(O_i-E_i)^2/E_i$
[0, 1.99)	137	155.3	2.1512
[2, 3.99)	98	91.6	.4411
[4, 5.99)	52	54.1	.0831
[6, 7.99)	36	32.0	.5000
[8, 9.99)	22	18.8	.5314
[10, 11.99)	13	11.1	.3096
[12, 13.99)	9	6.6	.8727
[14, 15.99)	6	3.9	1.1308
[16, ∞)	6	5.6	.0280
Total	379	379.0	$\chi_0^2 = 6.0479$

The degrees of freedom (k-s-1) for the NAS data is 9-1-1 = 7. At $\alpha = .05$, $\chi^2_{.05,7}$ from the chi-square tables is 14.1. Since $\chi_0^2 = 6.0479$, H_0 is accepted. Thus, the NAS login interarrival times are exponentially distributed.

The chi-square test verifies the logins to both the CDC and the NAS have exponentially distributed interarrival times; however, as highlighted earlier, these are the only systems with adequate data to make this analysis. Because of the lack of data, it is assumed all systems connected to the

CDS have exponential interarrival times. Once this assumption is made, there is still one key question to be answered. What is the estimated parameter λ for each of the systems?

Unfortunately, this parameter proved to be the most elusive of all the parameters to determine. The CDC and the NAS systems are the only ones keeping enough information to determine the number of logins per hour. Thus, these statistics are used to develop λ for the NAS and CDC. The λ parameter for the remainder of the systems was developed in one of two ways. The respective system managers were asked for an estimate of the number of logins. Some managers provided an estimate, but in other cases the managers were reluctant to make a guess. In those cases, the Table 1 requirements are used as the basis for the estimation. It is assumed the number of logins/hour to a system is 50% of the system requirements listed in Table 1 (6). Table 7 summarizes the parameters used for each supported system.

Statistical Model for Length of Login Session

The second parameter of interest for the CDS model is the length of a login and the associated distribution of the session length. As was the case with the previous parameter, the development of a statistical model for a session length is complicated by the lack of readily available information. The data necessary to determine the average session length is available on most systems, but calculating the underlying distribution is more difficult. However, the Modcomp system

maintained a daily accounting of the connect hours and the number of sessions associated with this connect time. A review of six months of data is used to determine the distribution of the login duration.

After collecting the raw data a histogram is formed to identify the statistical distribution. Table 5 summarizes the raw data collected for this analysis. Upon examining the histogram in Figure 5, it appears the histogram resembles a normal distribution with one exception. A normal distribution extends to positive and negative infinity; however, it is not possible for a login period to be negative. Therefore, the Modcomp session duration must be truncated on the left at zero. As with the previous statistical model, it is important to determine the parameters for the assumed distribution, and test this assumption using the chi-square test.

Table 5. Session Length Data

Length of Login (minutes)	Number of Observations	Length of Login (minutes)	Number of Observations
0-2	0	28-30	3
2-4	0	30-32	4
4-6	1	32-34	4
6-8	2	34-36	1
8-10	3	36-38	3
10-12	8	38-40	3
12-14	6	40-42	1
14-16	12	42-44	0
16-18	15	44-46	0
18-20	12	46-48	1
20-22	10	48-50	0
22-24	10	50-52	1
24-26	6	52-54	1
26-28	7		
		Total	114

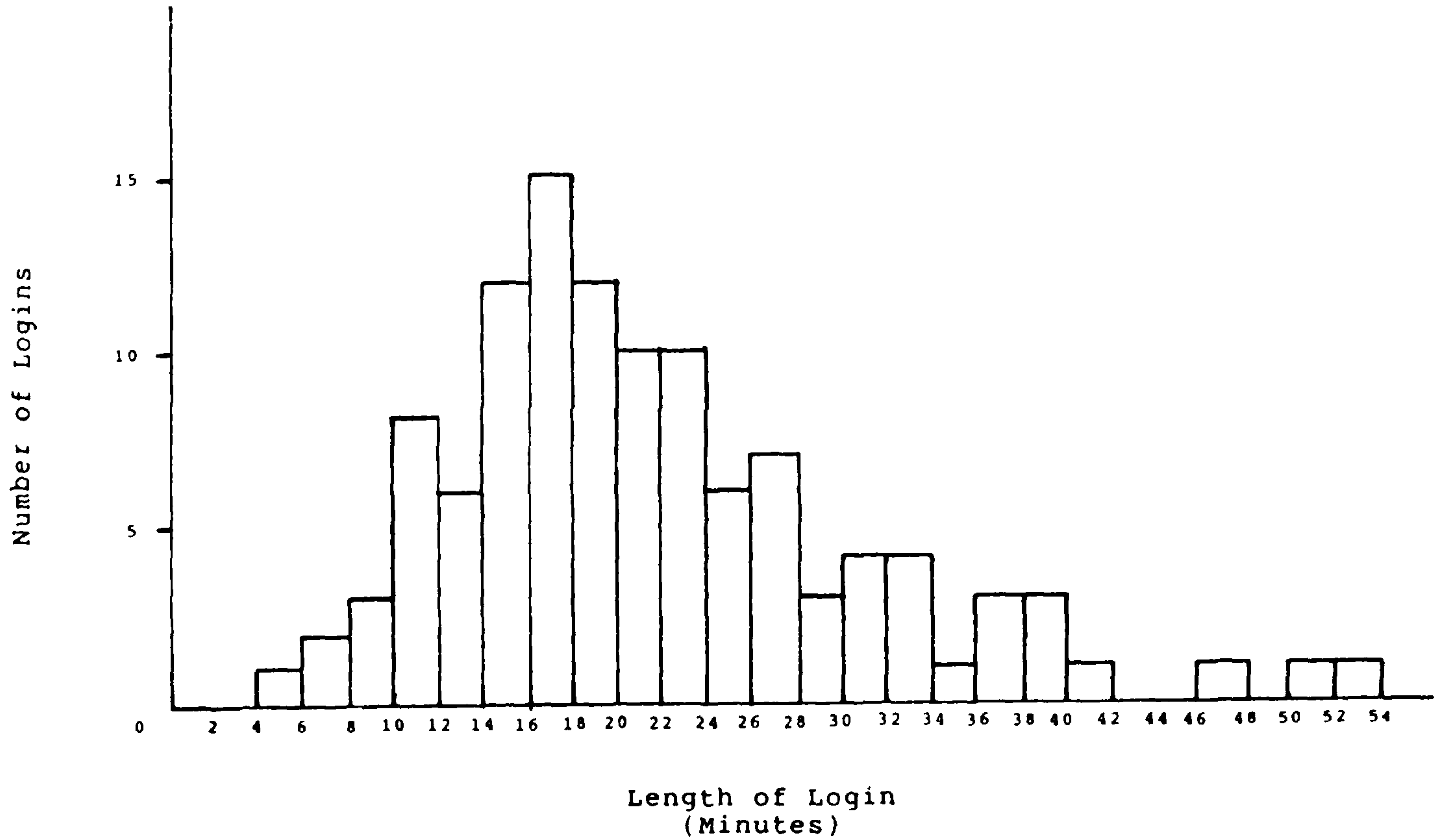


Figure 5. Histogram of the Modcomp Session Lengths

A normal distribution has a pdf given by

$$f(x) = (1/\sigma(2\pi)^{1/2})\ \exp[-1/2((x-u)/\sigma)^2],\ -\infty < x < \infty \qquad (6)$$

where u is the mean and σ^2 is the variance. In order to proceed with the analysis of the hypothesized distribution, u and σ^2 must be estimated from the collected data. As discussed previously, if there are n observations then the estimated mean u equals $\overline{X}$ (see Eq 2). The estimated variance σ^2 equals S^2, where S^2 is defined by

$$S^2 = (\sum_{i=1}^{n} X_i^2 - n\overline{X}^2)/(n-1) \qquad (7)$$

Applying Eqs 2 and 7 to the data listed in Table 5, the estimates are u = 21.6265 minutes and σ^2 = 83.5384 (minutes)^2.

The normal cdf is also needed to complete the chi-square test. The cdf for the normal distribution is defined by

$$F(x) = 1/\sigma(2\pi)^{1/2} \int_{-\infty}^{x} \exp[-1/2((x-u)/\sigma)^2]\ dx \qquad (8)$$

According to Carson and Banks, "it is not possible to evaluate Eq 8 in closed form" (4:149). Many authors use a transformation of variables to evaluate Eq 8 (4:149, 9:423, 18:47-48). Letting $z = (x-u)/\sigma$,

$$F(x) = 1/(2\pi)^{1/2} \int_{-\infty}^{(x-u)/\sigma} \exp[-z^2/2]\ dz \qquad (9)$$

If $$\phi(z) = 1/(2\pi)^{1/2} \exp[-z^2/2] \tag{10}$$

$$F(x) = \int_{-\infty}^{(x-u)/\sigma} \phi(z)\,dz = \Phi((x-u)/\sigma) \tag{11}$$

The tables for the normal distribution are used to solve Eq 11. However, recall that the distribution for session length is truncated at x = 0. Thus, in this case the cdf is

$$F(x) = \begin{cases} c\Phi(-u/\sigma), & x \geq 0 \\ 0 \quad , & x < 0 \end{cases} \tag{12}$$

where $c = [1-\Phi(-u/\sigma)]^{-1}$

Therefore, for this particular case c is computed as follows

$$c = [1-\Phi(-21.6265/9.1399)]^{-1}$$

$$c = [1-\Phi(-2.366)]^{-1}$$

but $\Phi(-x) = 1-\Phi(x)$ and $\Phi(2.366) = .99086$

$$c = [.99086]^{-1} = 1.0092$$

Applying the above equations to the Modcomp analysis, the following hypotheses are formed:

H_0 : the Modcomp session length is normally distributed.

H_1 : the Modcomp session length is not normally distributed.

Table 6 summarizes the chi-square test results for the Modcomp.

Table 6. Modcomp Chi-Square Goodness-Of-Fit Test Results

Interval (Minutes)	Observed Frequency, O_i	Expected Frequency, E_i	$(O_i-E_i)^2/E_i$
(0, 9.99)	6	10.6900	2.0576
(10, 11.99)	8	5.1642	1.5572
(12, 13.99)	6	6.4866	.0365
(14, 15.99)	12	7.3986	2.8617
(16, 17.99)	12	8.8578	1.1147
(18, 19.99)	12	9.6672	.5629
(20, 21.99)	10	10.0548	.0003
(22, 23.99)	10	9.9636	.0001
(24, 25.99)	9	9.4164	.0184
(26, 27.99)	7	8.4702	.2552
(28, 31.99)	7	12.9732	2.7502
(32, 35.99)	5	8.1738	1.2324
(36, ∞)	10	6.6836	1.6456
Total	114	114.0000	14.0928

Since the mean and standard deviation are estimated from the original data, s = 2. Therefore, the degrees of freedom (k-s-1) for the analysis is 13-2-1 = 10. At $\alpha = .05$, $\chi^2_{.05,10}$ from the chi-square tables is 18.3, and $\chi^2_0 = 14.0928$. Thus, $\chi^2_0 < \chi^2_{.05,10}$ and H_0 is accepted.

The chi-square test verifies the session duration for the Modcomp system is normally distributed. Based on this analysis, it is assumed the session or login duration for all supported systems follow a normal distribution and all available information is used to develop the mean and variance for each system. The results of this analysis are summarized in Table 7.

Statistical Model for the Amount of Data Transferred During a Session

The third parameter used to characterize the workload is the amount of data transferred during a session. Although the data transfer includes special traffic such as electronic mail or file transfers, these types of traffic are handled as separate inputs to the model and they are discussed in the next section.

The CDC was the only system having collected data applicable to this statistical model, but Appendix D of the CDS specifications includes an estimation of the volume of traffic for most of the supported systems. This estimation includes a minimum, a maximum, and an average amount of data transferred for a given system. However, there is no indication of what the underlying distribution is, so the distribution is assumed to be normal. This assumption is reinforced by the Central Limit Theorem. Simply stated, it says given enough observations of a random variable, the distribution of the mean tends to be normal (18:194). Based on this assumption, the CDS specification estimates are treated as follows. The average volume is the mean and the maximum volume is assumed to be within two standard deviations. Table 7 summarizes the actual numbers used for the different systems.

Other Input Parameters

As previously mentioned, file transfers and electronic mail are treated as separate inputs to the model. The

electronic mail (EM) is divided into two categories: local EM and DDN EM. It is assumed each interactive or Telnet session has an equal probability of sending and receiving 0 to 2 local EM messages and 0 to 2 DDN mail messages. Additionally, it is assumed the CDS processes 40 DDN EM messages per hour when serving as a gateway to other ASD systems. This figure was arrived at after a review of the AMS EM traffic and the Area B communications study. Based on the communications study, each EM message is assumed to be normally distributed with an average length of 4K bytes and a standard deviation of 2K bytes (3:Appendix F).

The file transfers, both FTPs and transfers to the CFS, are listed as requirements in Table 1. With the exception of the CFS, there are no available statistics to derive the number of transfers between systems; therefore, Table 1 is used to develop this parameter. The number of file transfers is assumed to be 50% of those listed in Table 1 with a minimum of one per hour. Table 8 summarizes this analysis.

When reviewing Table 8, keep the following principles in mind. The DDN transfers are evenly split between files sent to the DDN world and files received from the DDN world. The CFS transfers represent transfers to the CFS. The remainder of the entries represent the source of the file transfers. These same syscems act as the sinks for these transfers.

The communications study is once again used to determine the size of the file transfers. Based on a review of this document, the file transfers are modeled as having a mean of 200k bytes with a standard deviation of 250k bytes

(3:Appendix F). Lastly, the CFS transfer size is based on an analysis of current CFS statistics.

Summary of Input Parameter Assumptions

A term closely associated with computer programming is "garbage-in garbage-out." This also holds true for simulation models, for this reason much of the thesis effort concentrates on developing the required input parameters. However, despite this effort it is still necessary to make the following assumptions:

1. The interarrival times for logins, file transfers, and DDN EM bound for other ASD systems are exponentially distributed.
2. When there are no statistics or estimations of the number of logins or file transfers per hour, the parameter is derived from Table 1. It is assumed the number of logins/hour to a specific system is 50% of the number of users listed in Table 1.
3. All logins to the CDS have an equal probability of sending and receiving 0 to 2 EM messages both locally and via the DDN.
4. The session length durations are normally distributed.
5. The amount of data transferred during a session, file transfers, and EM messages is normally distributed.

Table 7. Summary of the Session Input Parameters

Type Of Traffic	Number of Logins/Hr	Session Length (Minutes)		Data Transferred in K Bytes	
		Mean	Std Dev	Mean	Std Dev
CDC Interactive	50.00	33.89	1.62	51.80	1.352
CDC Telnet	30.00	33.89	1.62	51.80	1.352
CDC HASP	5.00	33.89	1.62	66.67	25.000
NAS Interactive	56.00	40.92	6.45	27.79	1.693
NAS HASP	1.00	33.89	1.62	66.67	25.000
Modcomp 2400	1.66	17.83	5.53	375.00	17.080
Modcomp 9600	2.92	17.83	5.53	309.27	87.500
Modcomp 19.2	.42	17.83	5.53	80.00	66.670
VLCC Telnet	16.00	30.26	5.00	1510.00	793.000
SE Telnet	10.00	42.65	10.15	51.80	1.352
AMS Telnet	25.00	30.81	6.37	25.00	5.000
JLCF Telnet	16.00	42.65	10.15	51.80	1.352
AWC Telnet	7.00	42.65	10.15	51.80	1.352
INFOCEN Telnet	20.00	45.77	14.18	95.00	26.430
SEWS Interactive	2.00	33.89	1.62	51.80	1.352
DDN Telnet	40.00	45.77	14.18	95.00	26.430

Table 8. Summary of the File Transfer Parameters

System	Transfers/Hr	Data Transferred in K bytes	
		Mean	Std Dev
CDC	3.0	200	250
NAS	3.0	200	250
VLCC	1.0	200	250
SE	1.5	200	250
AMS	7.0	200	250
JLCF	2.0	200	250
AWC	1.5	200	250
INFOCEN	3.5	200	250
SEWS	1.0	200	250
LOCAL ETHERNET	2.5	200	250
DDN	3.5	200	250
CFS	2.0	814	200

III. The CDS Simulation Model

Introduction

"Simulation offers a satisfactory evaluation technique for performance evaluation, be it selection evaluation, performance projection of not yet existing systems, or performance monitoring of systems in operation" (22:87). Although the installation of the CDS has begun, it is not fully operational. Therefore, the CDS model is basically a performance projection. "Within this context, the simulation model is a mathematical-logical representation of the system which can be exercised in an experimental fashion on a digital computer" (19:6). Before developing the model, two top-level decisions must be made: what level of detail is needed, and is the model discrete or continuous?

Since a simulation model is supposed to represent the system, it must include enough detail to insure the results are valid; however, it is not necessary to incorporate all the minute details of the system, because "a model is not only a substitute for a system, it is also a simplification of the system" (4:9). Keeping this objective in mind, the CDS model is developed at the macro level.

While most systems are not entirely discrete or continuous, they can be predominately classified as one or the other (4:7). Haigh feels that "discrete-event simulation is an effective tool for analyzing the behavior of large communication networks" (14:177). Since the CDS is a communications network, the model uses discrete-event

simulation. Based on these top-level decisions, the following sections describe the development of the CDS model, including a general description of the six main modules of the model.

Simulation Language for Alternative Modeling (SLAM) II

The CDS model is developed in SLAM II, a FORTRAN-based simulation language. The SLAM II code is executed on the Pyramid 98Xe super-mini computer. SLAM II was selected for two reasons. First, "its process-oriented statements are suitable for modeling a computer communication network" (13:23-24). Second, the CDS deliverables include a SLAM-based capacity planning tool. Therefore, if the simulation model is developed in SLAM, it will be compatible with the capacity planning product.

SLAM incorporates the best features of simulation languages and general-purpose languages. It has a reduced instruction set allowing easy programming of certain network problems, but it also includes a feature allowing a designer to write FORTRAN subroutines. This feature gives the designer the ability to customize a model. Since the CDS simulation model only uses the network view, the following discussion addresses the network view.

The network view is analogous to flowcharting. The network modeler starts by developing a graphical representation of the flow of entities through the system. An entity (packet, message, etc.) is the basic unit of the simulation model. SLAM specifies network symbols that represent the different segments of a network. In the SLAM

world, entities flow through the system and they are processed at various points called "nodes." SLAM has twenty-two types of nodes providing such functions as "entering or exiting the system, seizing or freeing resources, changing variables, collecting statistics, and starting or stopping entity flow (17:89). The nodes are connected by branches called "activities." These branches define the routing of entities within the system. Entities may be assigned unique characteristics called "attributes." These attributes are used to control the processing of the entity as it traverses the system, and the attributes are used to collect statistics on the entities (17:89-90).

An important step in any simulation project is the analysis of the output. SLAM provides the following output reports: the input listing, echo report, trace report, and the SLAM II Summary Report. The first three reports are useful during the debugging and verification phases of the model development. The SLAM II Summary Report provides statistical results generated by the simulation. The report includes statistics collected on files, activities, and resources. The SLAM II Summary Report provides the necessary data to predict system performance or compare alternate designs (19:278-282, 13:25).

Model Assumptions

As stated in the introduction to this chapter, a simulation model should be a simplification of the real system. In order to accomplish this objective, certain

simplifying assumptions are usually made and the CDS model is no exception. In addition to the input assumptions listed in Chapter 2, the following assumptions are made:

1. All interactive data transfers are divided into packets with a fixed length of 128 bytes. This length is chosen because it is consistent with the X.25 standard. Additionally, the file transfers are segmented into 10-packet blocks.

2. For an interactive session, data transfers are bidirectional; however, the transfers are not equal in both directions. With the users terminal as the reference point, for every packet transmitted by the user there are ten packets received. This 10-to-1 ratio is selected for two reasons. First, an analysis of the AMS Local Area Networks (LANs) verified this ratio. Second, Randy Barker, designer of the CDS, stated a study he did for NCR also came to the same conclusion (5).

3. For interactive sessions, the transmitted packets are uniformly distributed across the duration of the login.

4. The CDS ethernets are modeled as queues with a deterministic service rate. Hughes and Li verified that this is an acceptable assumption when "loading is not high (i.e. 50-70%)" (15:215).

5. 25% of all local Telnet logins are multiple sessions (connections established to more than one system) (6).

6. All file transfers and local EM occur as a part of an interactive session.

7. The CDS components serve the packets on a first-come first-serve (FCFS) basis.

8. Each component has an infinite queue length.

9. Based on conversations with the CDS designers, it is assumed the UAPs and CIPs can process 160 packets/sec and the NAPs can process 320 packets/sec (5).

Approach Taken in Building the Model

The CDS model is built using an extended queueing approach. Sauer and MacNair claim a basic queueing network "consists of a set of jobs which visit queues and request service from the servers at those queues" (20:34). This concept is appropriate for very simple networks, but there is a need for extensions to handle situations such as seizing and releasing resources, allowing one job to spawn other independent jobs, and using variables to route and process jobs. All of these extensions are necessary to represent the CDS. Using these extensions the following approach is taken to building the CDS model.

The main emphasis in the CDS model is on the login or interactive session. Entities are created that represent an attempt to log into the CDS. Each created entity includes five important features: system requesting service from, input baud rate, output baud rate, session duration, and amount of data transferred. The input and output baud rates are used to seize available data communication links. If either an input or output line is not available, the entity

is balked away otherwise the resources are reserved and the entity continues through the model.

The entity is then split into packets. Since the packet length is 128 bytes, the amount of data transferred is divided by 128. This calculation determines the number of packets generated during the session. Recalling the 10:1 receive-to-transmit ratio, 1/11th of the packets are assigned as transmit packets. These packets are uniformly spread out over the length of the interactive session. When a transmitted packet reaches the destination system, the equivalent of ten packets is transmitted back to the user. This sequence of events is repeated until a logout message is transmitted by the user. When the logout message reaches the destination system, the reserved resources are released.

There is one exception to the above scenerio. Any time there are bulk transfers such as file transfers or electronic mail, the 10:1 ratio does not apply. These types of transactions are basically one way; therefore, all the packets are sent in one direction.

The Simulation Model

Organization.

The CDS model is divided into six main modules. The interactive sessions and bulk transfers are created, necessary resources are reserved, the data traffic is split into packets, the packets are queued at the appropriate network component, they are routed from node to node, and the session is terminated. Based on these functions, the six

modules are: the create module, the reserve module, the packet module, the routing module, the network module, and the terminate module. The major activities performed by each module are shown in Figure 6. This modular approach allows the modeler to start out with a basic model, and add more details at a later date.

The Create Module.

As the name implies, this module generates the entities that traverse the network. There are two basic types of traffic generated by the create module - interactive sessions and bulk transfers. The more complex of the two is the interactive session entities. Since each system the CDS supports has different attributes, the sessions for each system are created separately. Each interactive creation follows the same basic flow; therefore, a CDC login is used to explain the details of the create module.

The sessions are created using an exponential distribution with the parameter as listed in Table 7. After creation, the first attribute assigned is the input baud rate. For the CDC, the only input rates are 1200 bps, 2400 bps, or 9600 bps. There are a total of 201 TTY input lines with the above mentioned baud rates. 48.3% of the lines are 1200 bps, 47.3% are 2400 bps, and 4.4% are 9600 bps. Thus, conditional ACTIVITY statements are used to determine the input baud rate. Attribute 2 is assigned the proper input rate identifier.

Next the output rate is determined i· much the same way as the input rate. For the interactive CDC sessions, 94% of

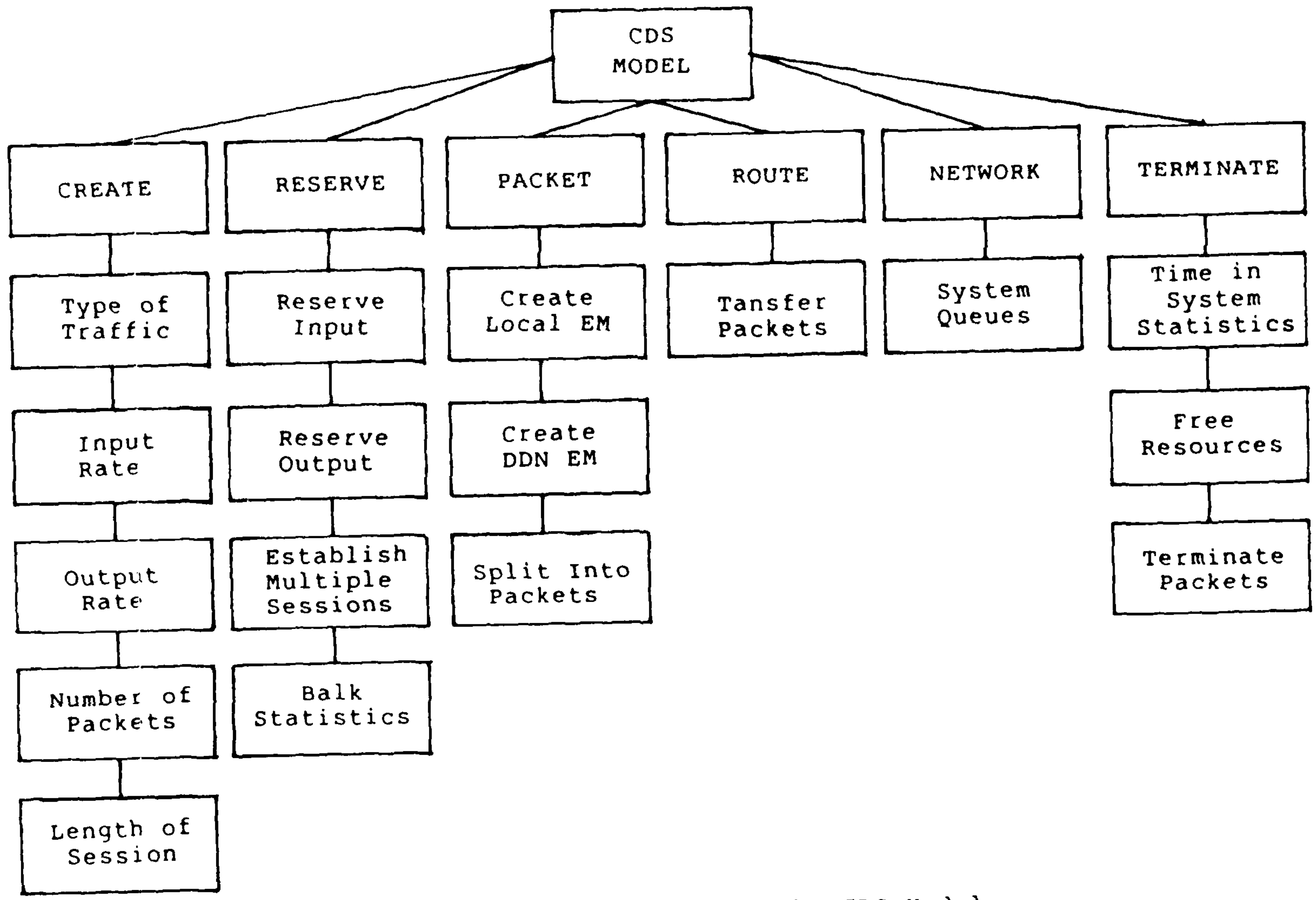


Figure 6. Main Modules of the CDS Model

the output line are X.25, the other 6% are autobaud TTY lines. Attribute 3 is assigned the output rate identifier and attribute 4 is assigned the traffic type identifier.

The final attributes assigned by the create module are the session duration and the amount of data transferred during the session. Both attributes are generated by normal distributions with the parameters as shown in Table 7. Attribute 5 is assigned the duration of the connection and attribute 6 is assigned the total number of packets transferred during the session. Finally, the entity is routed to the reserve module.

There is one exception to the above interactive scenario, a multiple login. As stated in the assumptions, [illegible] of the Telnet logins are assumed to be multiple sessions. When a multiple session does occur, attribute 7 is set to one.

The bulk transfers go through basically the same steps as the interactive sessions except a session duration (attribute 5) is not assigned. Additionally, the source and destination indicators are assigned to attributes 9 and 12 respectively and the entity is sent to the routing module.

The Reserve Module.

The reserve module seizes and frees TTY and X.25 lines that serve as both input and output lines. Three types of entities use this module: those needing to reserve both the input and output lines, those needing to reserve an output line only, and multiple logins.

When both an input and output line is needed, this module first checks for the availability of the input line. If the proper type of input line is available, the available resource number and the input baud rate are used to determine the input CIP. This information in turn is used to determine the input UAP. Attribute 8 is assigned the input CIP number and attribute 9 is assigned the input UAP number. If an input line is not available, the entity is balked away and statistics are collected on the number of balks by input rate.

Next the availablity of an output line is checked. The process is the same as the above procedure except the output CIP number is assigned to attribute 11 and the output UAP to attribute 12. If an output balk does occur, not only are the statistics collected, but the input resources are also released.

In some cases the input of the CDS may be via the DDN or an ethernet; however, the connection to the requested system may still be a TTY or X.25 line. When this occurs an output link must be reserved and the entity bypasses the input portion of the reserve module and goes directly to the output block.

If a multiple login does occur, there is no need to reserve resources but the input link must be assigned to a connection already being utilized. This segment of the model checks for an appropriate connection and piggybacks on the connection. Thus, attribute 8 equals the input CIP and attribute 9 equals the input UAP.

Regardless of the type of entity entering the reserve module, there are only two ways to exit this module. Either all the resources are available, in which case the entity exits to the packet module, or the entity is terminated due to a balk.

The Packet Module

The packet module performs two major functions. First, it creates the EM associated with certain interactive sessions. Second, it takes the sessions and breaks them into packets for transmission.

The original entity is split into three entities. One entity represents the original interactive session, the second one represents the local mail, and the last entity represents the DDN mail. The module uses a uniform distribution to determine the number of local and DDN mail messages. In both cases, there is an equal probability of creating 0 to 2 messages. If the number of mail messages is 0, the respective entity is destroyed. The number of messages associated with the session is assigned to attribute 13. An UNBATCH node uses attribute 13 to split out the necessary mail messages. Then a normal distribution is used to determine the number of packets associated with each message. This number is assigned to attribute 6. For local mail messages, the remainder of the attributes retain the values of the original interactive session, except for attribute 4. Attribute 4 is assigned the local EM indicator. For DDN messages, attribute 12 is assigned the DDN output indicator, attribute 4 the DDN EM indicator,and the remainder

of the attributes remain the same. Then the entire mail message is delayed for some period of time between 0 seconds and the length of the login session. After the delay, the messages are sent to the routing module.

Once the mail messages are created, the original session entity must be processed. Attribute 6 contains the number of packets associated with the session. Recall that for an interactive session there is a 10:1 ratio between received and transmitted data. Thus, attribute 6 is multiplied by 1/11 to determine the transmitted packets, and this number is assigned to attribute 6. Attribute 14 is set to 1 to indicate transmitted packets. The entity is routeed to an UNBATCH node and attribute 6 is used to split out the transmit packets. Once the transmitted packets are unbatched, attribute 6 is set to 1. As with the mail messages, each packet is delayed for some period of time between 0 seconds and the session length, and then each packet is sent to the routing module. This delay effectively spreads the data transfer uniformly across the session length.

The Routing Module.

The routing module transfers the packets from network component to network component. The module uses a combination of seven different attributes to route the packets through the network. The only attribute modified by this module is attribute 16. Attribute 16 indicates where a packet was last offered service. The routing module uses attribute 16, plus the source, the destination, type of

traffic, and attribute 14 to transfer the packets from node to node.

The Network Module.

The network module is the actual queues representing the components of the CDS. The network uses a queue to represent the CIPs, UAPs, ethernets, NAPS, and the DDN input and output links. The activity time of each component is dependent upon two things: the number of packets per entity (attribute 6) and the per packet service rate of the given component. The activity time for a given entity is determined by multiplying attribute 6 by the component's per packet service rate.

The Termination Module.

The final module performs three major activities. First, it gathers statistics on the packet delay on certain packets bound for the DDN world. Second, it determines whether the packet was a transmit or receive packet. If the entity is a transmit packet (attribute 14 = 1), this module creates a receive packet. This procedure includes setting attribute 14 equal to 2, attribute 6 equal to 10, and sending the entity to the routing module. If the entity needs no more processing, the resources are freed and the entity is terminated. Finally, Figure 7 shows the general flow of an interactive session through the various modules and Table 9 summarizes all the attributes used in the CDS model.

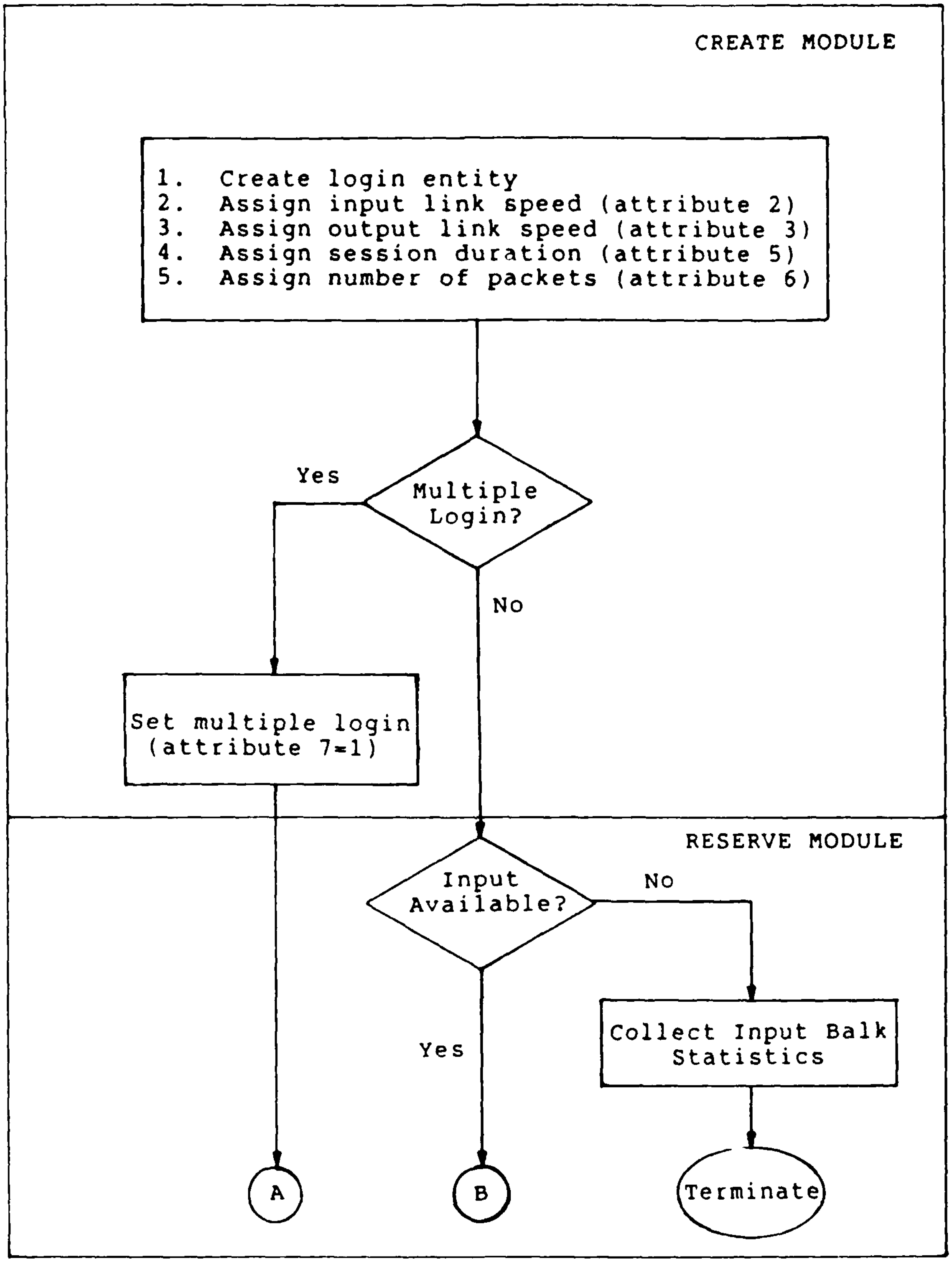


Figure 7. General Flow of an Interactive Session Through the CDS Model

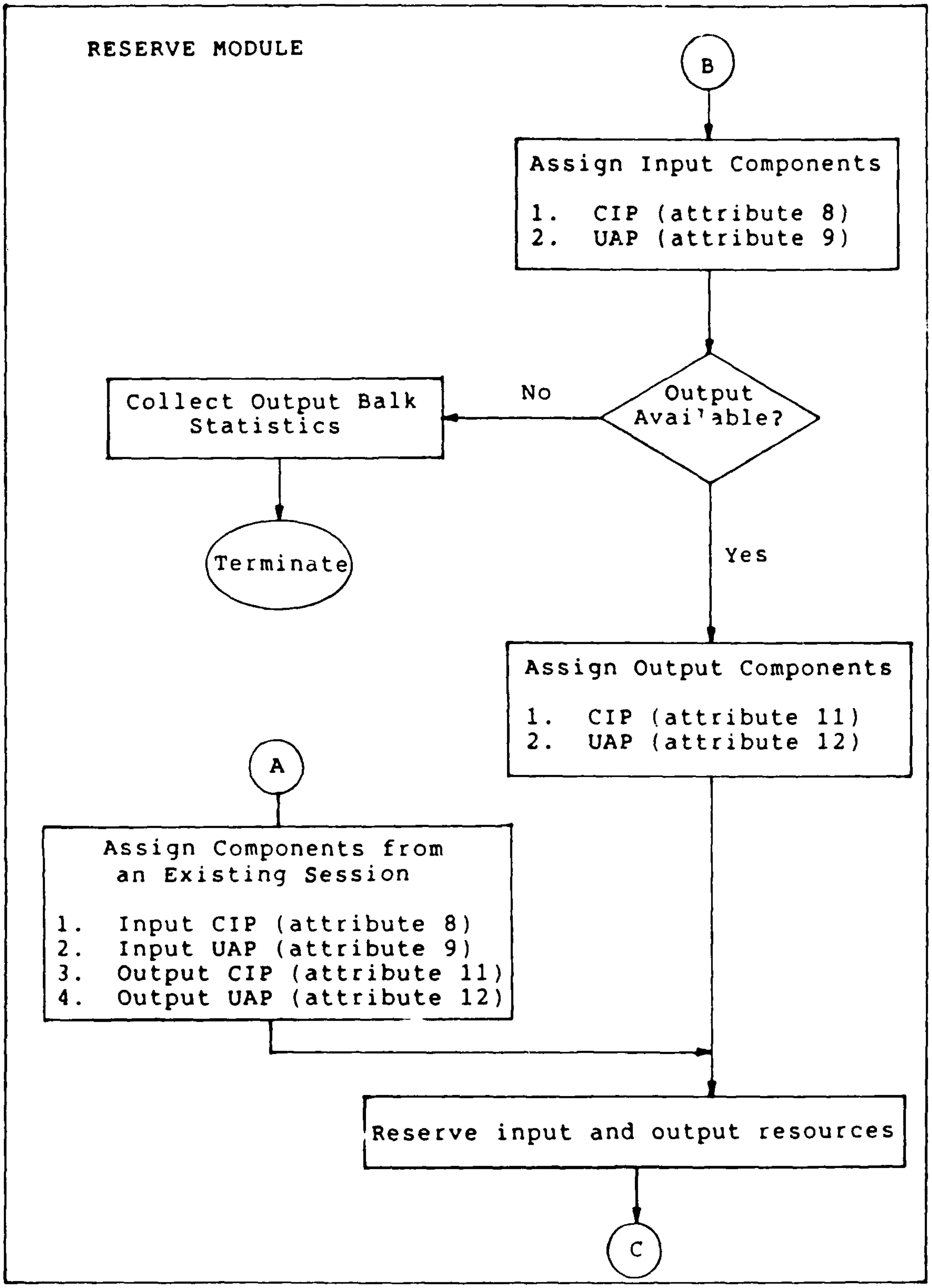


Figure 7. General Flow of an Interactive Session Through the CDS Model (continued)

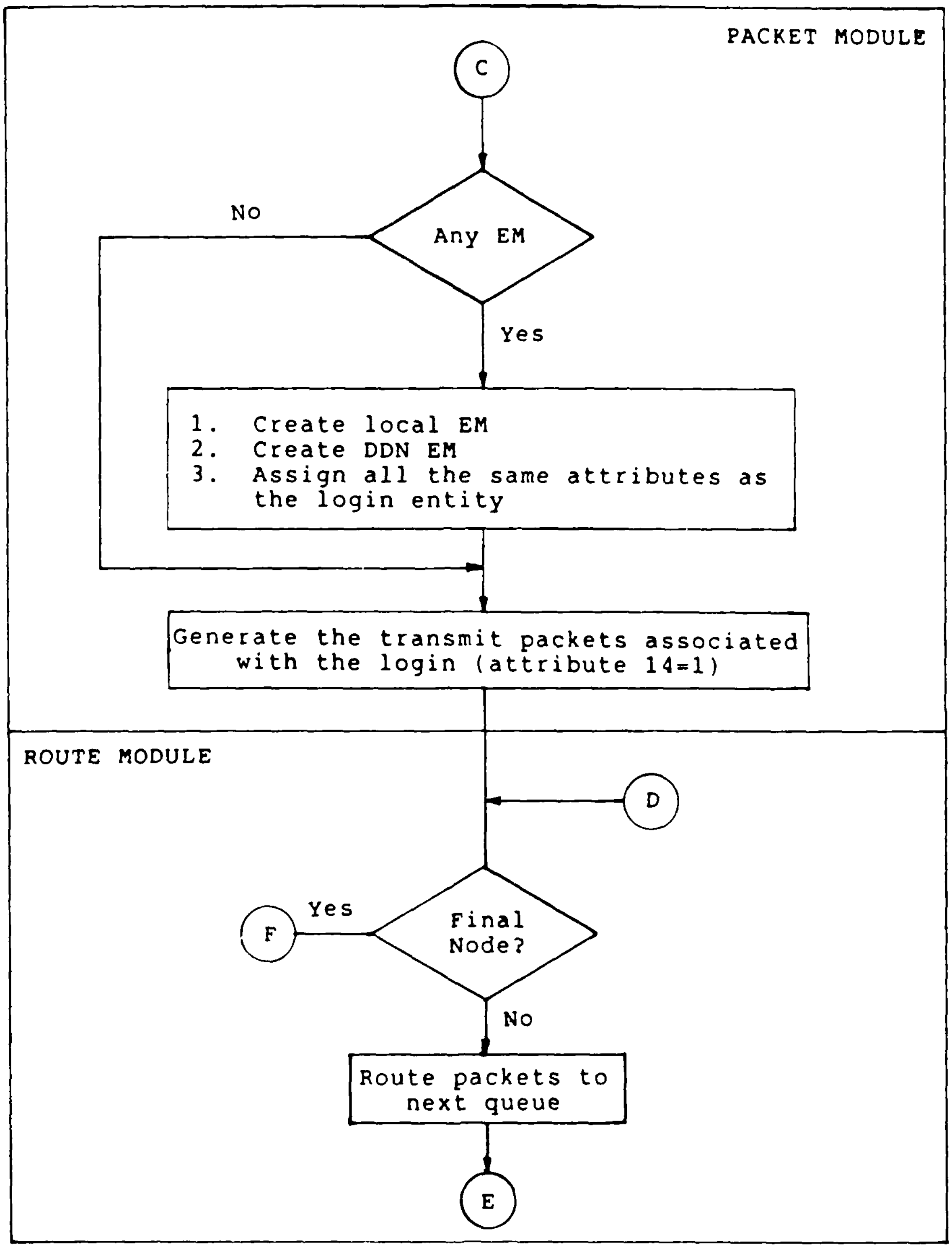


Figure 7. General Flow of an Interactive Session Through the CDS Model (continued)

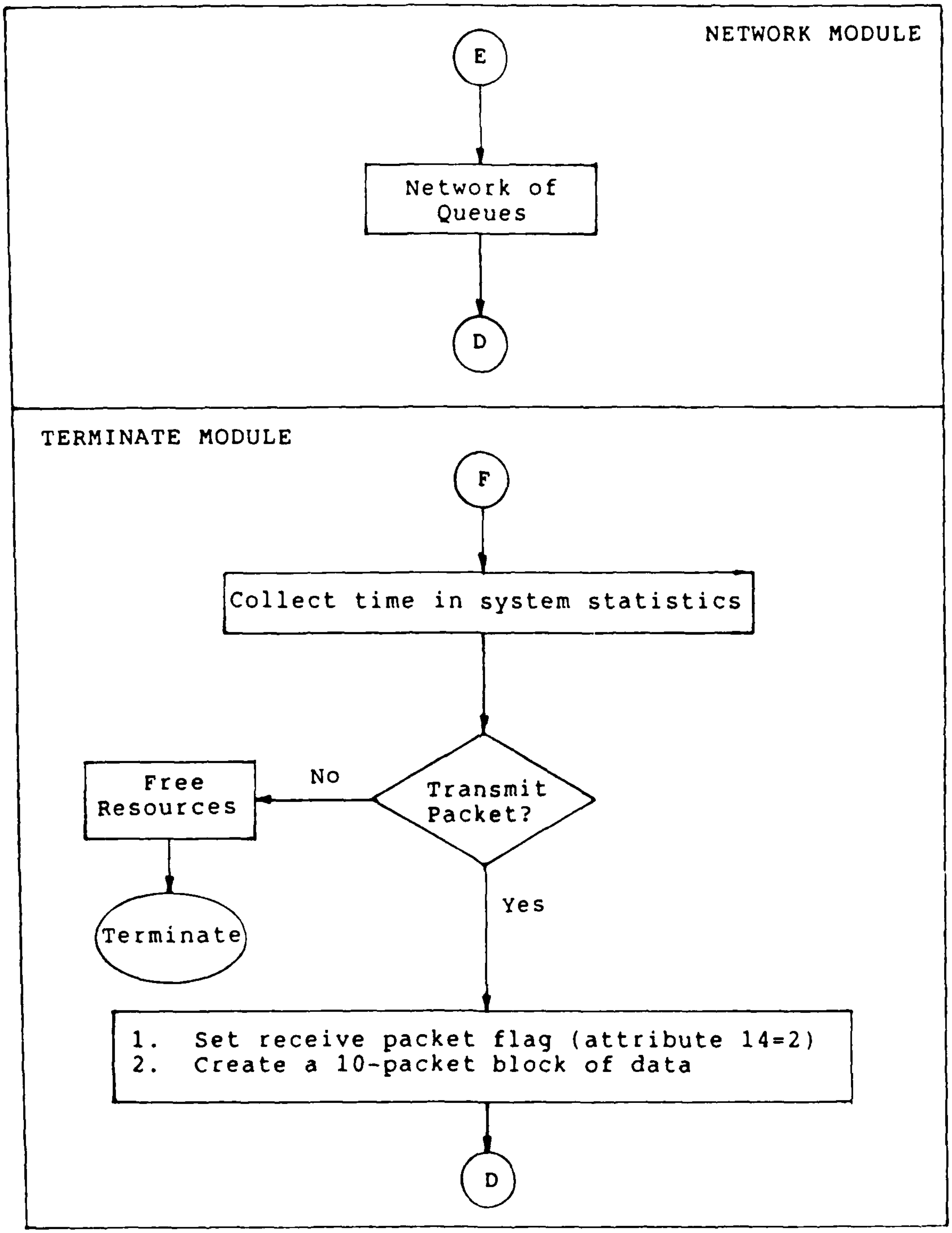


Figure 7. General Flow of an Interactive Session Through the CDS Model (continued)

Table 9. Packet Attribute Definitions

Attribute	Definitions
1	Creation Time
2	Input Link Speed
3	Output Link Speed
4	Type of Traffic
5	Login Duration
6	Number of Packets
7	Multiple Login Flag
8	Input CIP
9	Input UAP/NAP/Ethernet
10	Temporary Storage
11	Output CIP
12	Output UAP/NAP/Ethernet
13	Unbatch Count for EM
14	DDN Link
15	Not Used
16	Last Node Flag

Model Verification and Validation

Shannon divides the evaluation of simulations into three categories: verification, validation, and problem analysis (23:30). The problem analysis is very involved and Chapter 4 is entirely devoted to this analysis, but before using the model to draw conclusions, the modeler must have faith that the model is valid. This is the role of model verification and validation.

The verification process insures the model behaves as the modeler intended. The verification of the CDS model is a three step process. First, the coding is always double checked to insure there are no errors in translating the code. The SLAM Summary Report helps in this area. Any time there is an error in the input code, the output listing of the SLAM Summary Report lists any errors in the input code. Second, the SLAM trace feature is used to check the

progression of entities. The printed trace report is hand checked to insure specific packets are flowing through the intended components. Once the CDS model passes both of the above tests, there is a high degree of confidence that the model is functioning as intended. The final test is to see if the model produces reasonable results. Since the CDS is still being installed, there is no real system to compare the simulated results against. However, the CDS committee members and the CDS designers have many years of experience in the area of computer systems. This expertise is used to check the reasonableness of the model. The results of the simulation were discussed with these experts, and it was felt that the results were believable (5,6).

Normally, the first step in the validation process involves examining the input-output transformations. Carson and Banks insist that a "necessary condition for the validation of input-output transformations is that some version of the system under study exists" (4:387). The CDS is not operational; therefore, the validation process is limited to two steps. First, where possible the model assumptions are validated or tested for goodness of fit. All of Chapter Two is devoted to this task. Second, the model is tested for face validity. Sensitivity analysis is used to test face validity of the model. The CDC logins, NAS logins, and the DDN traffic are varied $\pm$ 10% and the reaction of the model is observed. Since the results did not change greatly over this range, the CDS model is assumed to be valid.

IV. OUTPUT ANALYSIS OF THE CDS MODEL

Introduction

Once the model is validated, it is used to draw some conclusions about the network. Before exercising the model, the exact design of the experiment must be established and the number of runs needed to produce the desired information must be determined. Shannon calls these two steps of the simulation process the strategic planning stage and the tactical planning stage (23:23). The next two sections of this chapter discuss the design of the experiment. With this framework established, the remainder of the chapter analyzes the output of the model, including a discussion of the findings.

Experiment Design

In order to design an experiment, it is important to first determine what questions the model is suppose to answer. There are two reasons for the design of this model. First, to determine if the present CDS design can support the defined Phase I requirements, and second, to determine the effects on the system as the DDN workload is varied from the baseline configuration up to a 200% increase. Thus, the experiments must be designed to supply enough information to answer these rather broad questions.

The Baseline Model.

To answer the first question, it must be determined what constitutes supporting the CDS workload. Since the main function of the CDS is to act as the central user interface

to the ISTC, it is imperative that the system does not introduce an unacceptable delay to the local interactive sessions. According to Dr. Jurick, the human attention span when interfacing with a local system is approximately 2 seconds (16). The CDS must not introduce a delay that exceeds this limit when combined with the delays from the other communication links in the connection. Therefore, the baseline model must determine the delay incurred by a packet traversing the network. The major problem with determining this time is there are many different paths through the CDS network with some paths slower than others. Averaging the packet paths together does not give a fair picture, because some paths use more components and certain components have a tendency to be more heavily utilized. This tendency comes from the fact that the polling of available resources always rotaries through the resources in the same order and always seizes the first available link. A trial run of the simulation verifies that this is occurring. Therefore, the time is gathered on the longest and most heavily utilized paths. If the design satisfies these slower links, then the other links surely meet the requirement. The paths in question would be a session coming into the CDS through a CIP and establishing a connection to an applications processor over an internetwork ethernet.

The second area to look at for the baseline design is the availability of the communication links. Questions to be answered are the number of each type of link that is normally available and if there is balking, how often does it occur?

The last area investigated on the baseline model is the utilization of the various components. This analysis not only includes a look at the average utilization, but also the deviation within a given run.

The DDN Investigation.

The purpose of this experiment is to look at the effects of increased DDN traffic on the CDS. Two constraints on this investigation have already been established: the range of interest is an increase in the DDN traffic of 0% to 200%, and the analysis must include DDN link speeds of 56 Kbps and 1.544 Mbps. With these parameters in mind, the experiment concentrates on two areas.

First, since the NAPs serve as the DDN gateways, the simulation investigates the average delay within the NAPs as the load increases. Second, the simulation output is used to establish some boundaries on the average delay incurred by a Telnet packet headed for the DDN. Once again there are many different paths through the CDS to the DDN world, so some specific links must be investigated to determine the upper and lower limits. The longest delay experienced by a DDN user will occur when the link is set up through the front end of the CDS; therefore, the upper limit is established by collecting statistics on the slower TTY connections to a DDN host. These TTY connections gain access to the CDS through a CIP. Conversely, a user establishing a DDN link from an internetwork ethernet connection is the shortest path (in terms of time) to the DDN. Thus, statistics on these types of connections are gathered to determine the lower limit.

Both of the above areas must be investigated over the range of consideration and at the two different DDN link speeds. However, before plunging into the analysis of the results, it is still necessary to determine how to terminate the individual runs and how many runs to make. This is Shannon's tactical planning stage.

Tactical Planning

Carson and Banks describe two "types of simulation with respect to output analysis: terminating or transient simulations and steady-state simulations" (4:412). A transient simulation starts with some well-defined starting conditions, and stops after some specified time or event. On the other hand, steady-state simulations are used to model nonterminating systems. Nonterminating systems run continuously or for an extended period of time. Communication systems are generally considered to be nonterminating systems (4:414).

In order to simulate a nonterminating system, the model must mimic the steady-state behavior of the system under study. To insure the simulation model accomplishes this objective, the modeler must specify the length of the run and the initial conditions. In the case of the CDS, the initial conditions are not defined; therefore, the simulation is started at time zero with an empty system and allowed to run for some period of time.

Starting with an empty system introduces start-up biasing. This start-up bias can be reduced by dividing the

simulation into two discrete phases. The first phase is the initialization phase and the second phase is the data collection phase. The initialization phase starts at time zero and runs until the collection phase. The purpose for this phase is to bring the system to the initial steady-state conditions. Before the data collection phase is started, the statistics are zeroed but all entities in the system remain. Thus, when the data collection phase starts, the system is not empty, but theoretically in a steady-state condition. The required statistics are now collected on the steady-state for some period of time and then the run is terminated (4:430, 19:43-45). This requires the modeler to make two decisions, when should the statistics be cleared and when should the run be terminated?

There are no magical equations to help determine the answer to these questions. The only way to arrive at these numbers is make some pilot runs to determine the required length of both phases. The following approach is used to insure the model reaches a steady-state condition. The simulation is run using four different periods for the initialization phase. The runs start out with an initialization phase of 15 minutes and it is increased by 15 minutes each run. In all cases the total length of the run is 5 hours and a SLAM Summary Report is printed every hour. At some stage the collected statistics should begin to approach some steady-state point. It is imperative to make the best use of the computer resources, so the data must be reviewed to see which combination of times yields

steady-state statistics, but does not use an excessive amount of computer time. Unfortunately, for the CDS model there were oscillations so the statistics did not completely settle down. Therefore, the run duration proved to be quite lengthy. The combination that was finally selected was a total run length of 4 hours, with the statistics cleared after the first 30 minutes. The above run length determination helps to reduce the variance within a run, but there is still the question of how many runs to make to insure the variance between runs is acceptable.

Unlike the previous analysis, there are some analytical methods to determine the number of runs to make. Once again the results of the pilot runs are used to make this determination. Before starting the analysis, the modeler must determine two parameters: the desired confidence interval and a specified accuracy criterion. For the CDS model, one of the parameters of interest is the mean delay incurred at the DDN gateways. A 95% confidence interval is specified for the mean delay, where $100(1-\alpha)\%$ is used to represent the confidence interval. Next the desired accuracy, ε, must be determined, Carson and Banks suggest an accuracy of at least (1-.95) (4:427). Since the DDN gateway delays are measured in tenths of seconds, $\varepsilon = .05$ seconds.

Initially 4 pilot runs are made. These runs are used to obtain an estimate, S^2, of the population variance. Now let R represent the total number of runs needed to attain the

confidence interval with the specified accuracy. In order to accomplish this the following relationship must hold

$$R \geq [(t_{\alpha/2,R-1}S)/\varepsilon]^2 \qquad (13)$$

For the pilot runs, $S = .0263$ seconds and $t_{.025,3} = 3.18$. Solving for Eq 13, $R \geq 2.804$ and the inequality is satisfied. Note if the number of runs is reduced to 3, $t_{.025,2} = 4.3$ and $R \geq 5.127$ and the inequality of Eq 13 is not satisfied; Thus, the analysis requires 4 runs.

Output Analysis

The Baseline Model.

Since CDS serves as the interface to the ISTC resources and the majority of the logins are interactive, the CDS must not introduce an unacceptable delay. The interactive traffic falls into one of two very broad classes. One category of connection is where the user connection is through a CIP and the connection on the application processor side is also through a CIP. The second connection involves a connection through the CIP and the computer connection is through an ethernet connection. An investigation of these connections reveals they introduce an average one-way delay of .1342 seconds and .1732 seconds respectively.

The second area of investigation on the baseline model deals with the availability of the input and output links. A key concern with any communications network is insuring users can gain access to system and not be turned away. The Phase I design provides an abundance of input and output links.

Repeated runs of the baseline model reveals that at no time was any login requested turned away because of lack of resources. Table 10 summarizes the average input/output utilization for the Phase I design.

Lastly, the baseline components are scanned to see the average utilization of the CIPs, UAPs, and NAPs. This review reveals that none of the components is more than 11% utilized. The utilization of the CIPs runs from a low of less than 1% usage to a high of 9% utilization. The average utilization on the UAPs is 7.8% and on the NAPs is 10.9%. While the utilization on any given component appears to be low, a word of caution is warranted. The nature of the traffic on the CDS is very bursty and this type of traffic pattern can cause temporary surges on the system components. These types of surges are evident within any given run. While some components exhibited very few fluctuations, the UAPs and NAPs all had bursts where the utilization reached as high as 45%.

<u>The DDN Analysis - 56 Kbps Links.</u>

The DDN analysis is designed to determine the CDS' response to increased DDN traffic. Recall from Chapter One, the original CDS design is based on 56 Kbps links to the DDN. To answer the question of whether or not the CDS can serve as the DDN gateway, the analysis begins with the previously described baseline model. All local traffic is held constant and the DDN traffic is increased in 25% increments. The analysis is continued until the DDN traffic has been

Table 10. Resource Utilization for Baseline Model

Resource (bps)	Maximum Available	Number Utilized	Percent Utilized	Maximum Number Utilized
Inputs				
1200	97	65.16	67.18%	87
2400	95	34.90	36.74%	60
4800	2	.41	20.50%	2
9600	9	3.64	40.44%	9
19.2 K	2	.81	40.50%	2
56 K	24	8.14	33.92%	16
Hasp 9.6 K	4	1.13	28.25%	4
Hasp 19.2 K	5	1.67	39.40%	5
Outputs				
CDC 2400	4	.97	24.25%	4
CDC 9600	4	.86	21.50%	4
CDC 9600 (X.25)	108	28.46	26.35%	45
CDC 19.2 K (Hasp)	10	2.96	59.20%	10
NAS 2400	12	3.04	25.33%	8
NAS 9600	12	3.05	25.41%	8
NAS 9600 (X.25)	68	14.12	20.76%	28
NAS 56 K	36	18.42	51.17%	31
NAS 19.2 K (Hasp)	1	.24	24.00%	1
Modcomp 19.2 K	12	1.44	12.00%	4
SEWS 9600	3	.88	29.33%	3

increased to a maximum of 200% of the baseline model. The analysis focuses on three main areas: the delay introduced by the DDN gateways, the minimum and maximum delay experienced by packets bound for the DDN, and the effect on the local traffic.

Figure 8 shows the average gateway delay incurred by a packet as it traverses the network. This delay ranges from a low of .1301 seconds to a high of .2373 seconds. From the figure it is obvious the delay is relatively linear with respect to the increase in DDN traffic. But the NAPs or DDN gateways only constitute one component in the link between a user and the DDN world. A more important question is what is the total delay of a packet destined for the DDN?

This is not an easy question because of the many different routes a packet can take through the CDS and some paths are going to take more time than others; therefore, a look at the longest and shortest path establishes the boundaries on the DDN-bound packets. Figure 9 shows these boundaries. These delays only apply to traffic addressed to or from the DDN, but what effect does the increased DDN traffic have on the local traffic?

The local traffic is divided into two general types: those logins whose input and output links are via a CIP and those logins whose input link is through a CIP and the output is via the internetwork ethernet. The delay experienced by the ethernet type connections is predominantly a result of the NAP delay of Figure 8. On the average, the NAP delay accounts for 65% of the delay on a CIP-to-ethernet

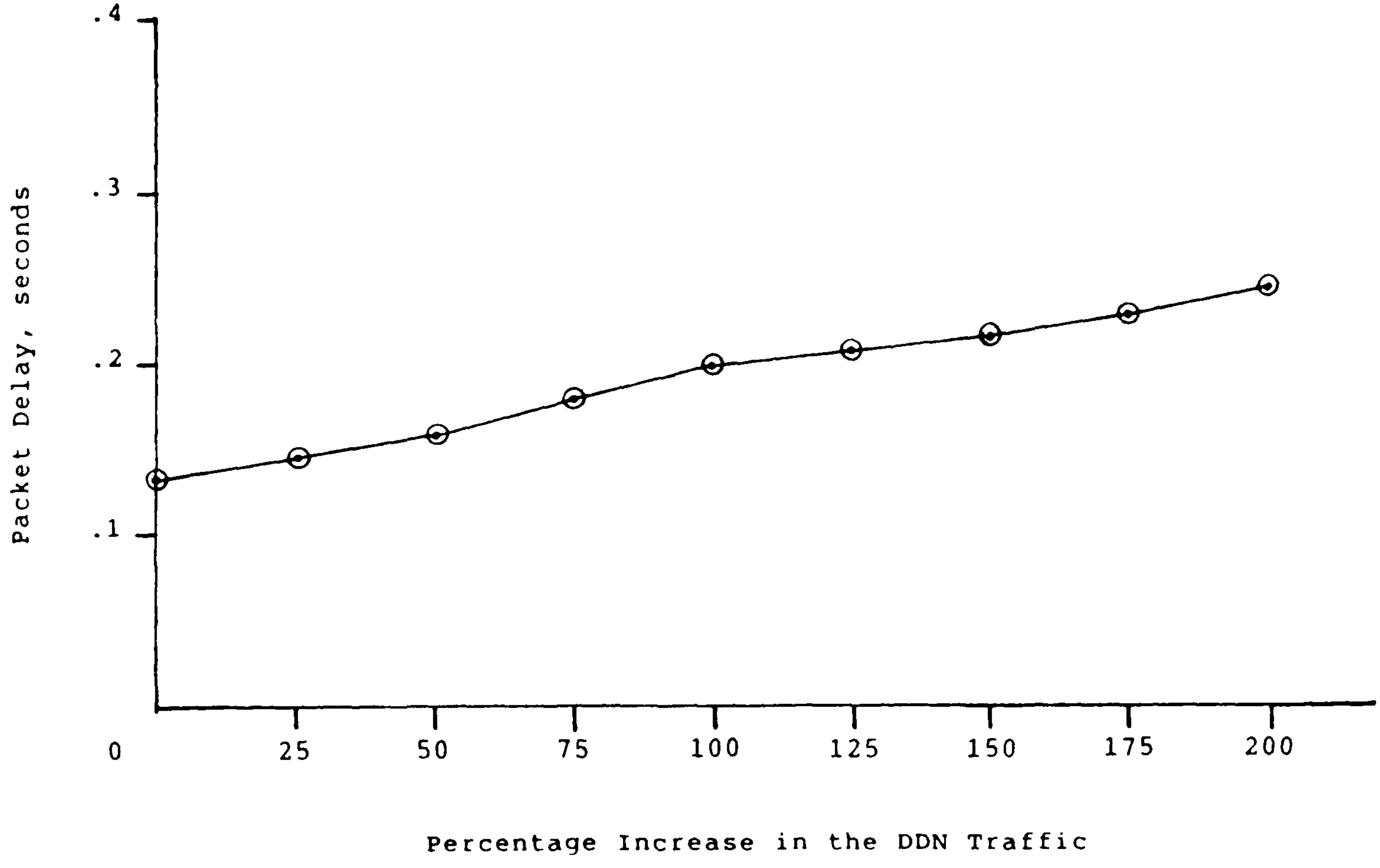


Figure 8. Average DDN Gateway Packet Delay - 56 Kbps Links

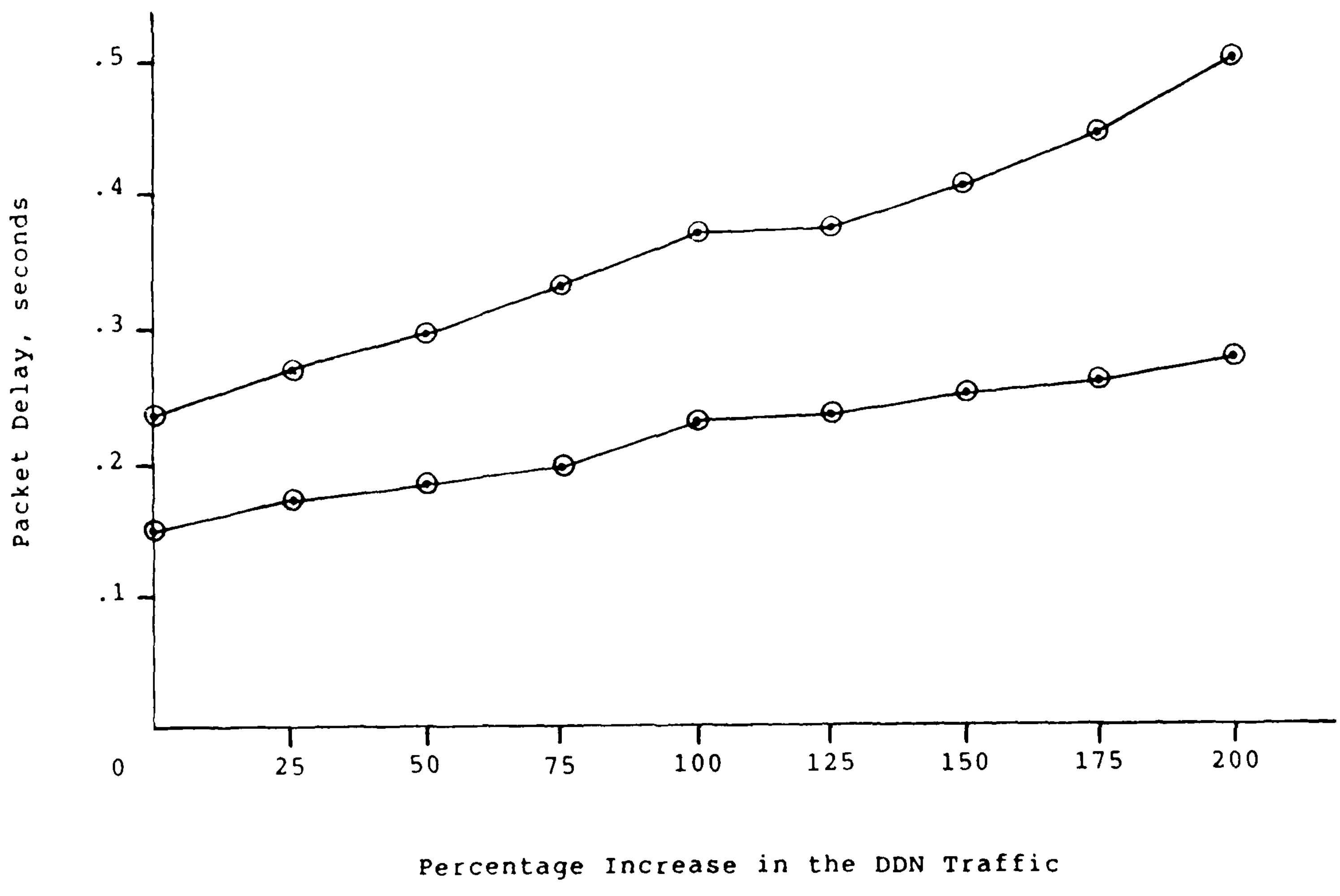


Figure 9. Minimum and Maximum DDN Packet Delay - 56 Kbps Links

connection; however, since the CIP-to-CIP connection does not use the NAPs, it exhibits a much shorter delay. Figure 10 shows the average delay for this type of login.

The key question to be answered by the above analysis is what is the maximum expected delay experienced by a DDN packet? While the upper boundary of Figure 9 tells part of the story, it does not give the complete picture. It is necessary to give a feel for the accuracy associated with these estimates. A commonly used technique for prescribing the upper and lower bounds for the estimates is by using confidence intervals. For this analysis a 95% confidence interval is assumed. To determine the confidence interval (CI) the following equation is used

$$CI = X + [(t_{\alpha/2,N-1})(S)(N-1)^{-1/2}] \qquad (14)$$

where N is the number of runs used to make the estimates, α the level of significance, S the standard deviation of the runs, and $t_{\alpha/2,N-1}$ the t distribution for the specified parameters. Thus, for the above analysis, $\alpha = .05$, $N = 4$, and S is determined using Eq 7. Table 11 summarizes this analysis.

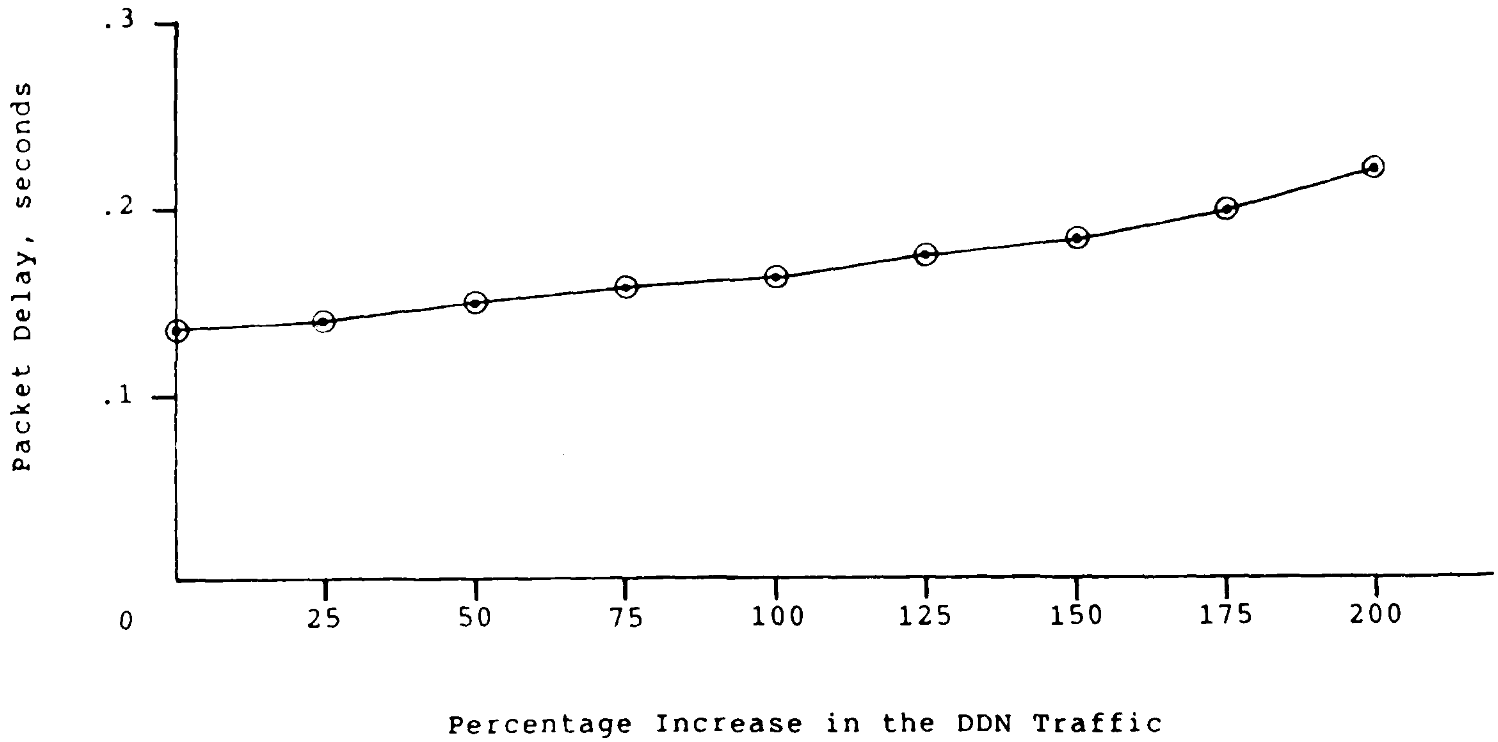


Figure 10. Packet Delay for CIP-to-CIP Connection - 56 Kbps Links

Table 11. Upper Bound With 95% Confidence Interval - 56K

Percent Increase	Upper Confidence Bound (seconds)
0	.2706
25	.3068
50	.3430
75	.3791
100	.4153
125	.4515
150	.4876
175	.5238
200	.5600

Finally, linear regression techniques are applied to the above numbers to find a straight line that best fits the Table 11 data points. Figure 11 shows the straight line upper limit for the DDN traffic. The equation for this line is

$$y = (.001446)x + .2706 \quad \text{(seconds)} \qquad (15)$$

Eq 15 gives the expected delay for a given increase in the DDN traffic. For example, if the DDN traffic is increased by 35%, x = 35 and y = .3212 seconds. A word of caution is in order, Eq 15 only applies for the range of study (0% - 200%).

<u>The DDN Analysis - T1 Links.</u>

The DDN analysis is repeated with the DDN link speed increased to 1.544 Mbps. Figures 12 through 14 show the NAP delay, the upper and lower boundaries, and the local traffic delays respectively. As is the case with 56 Kbps links, the NAP delays experienced when the DDN links are 1.544 Mbps are linear as the DDN traffic is increased (see Figure 12). These delays range from a low of .1361 seconds to a high of

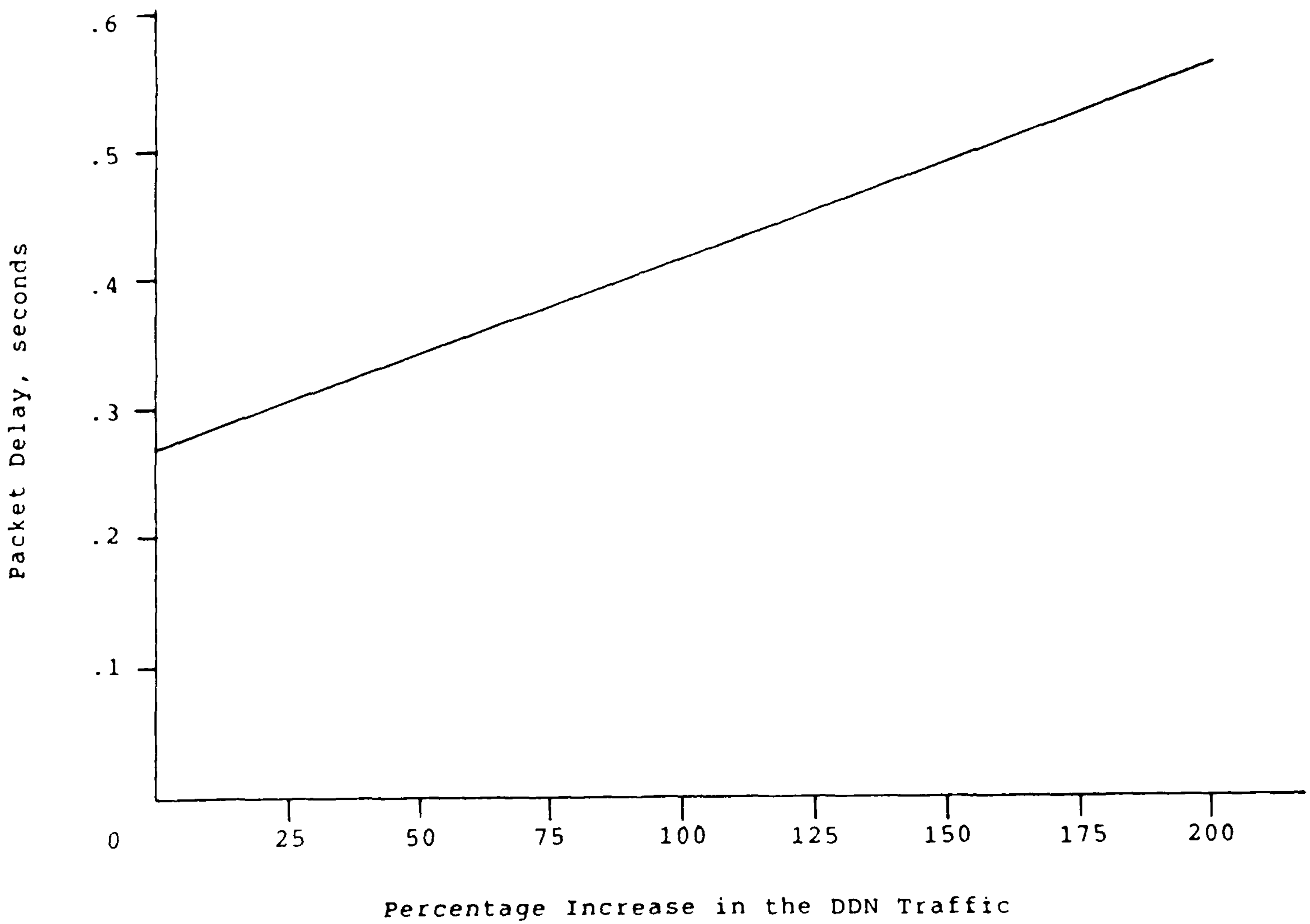


Figure 11. Maximum DDN Packet Delay with 95% Confidence Interval Added - 56Kbps Links

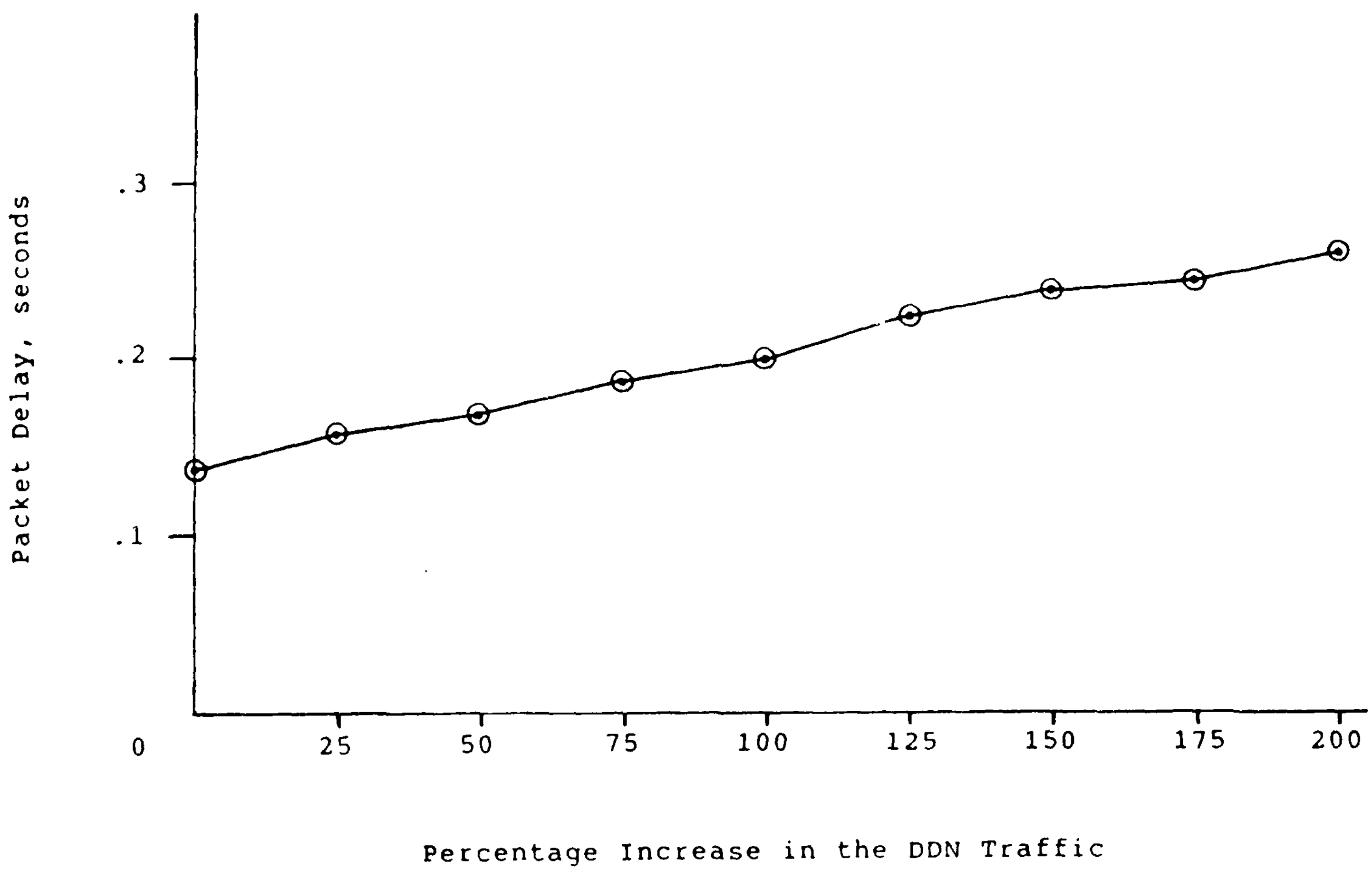


Figure 12. Average DDN Gateway Packet Delay - T1 Links

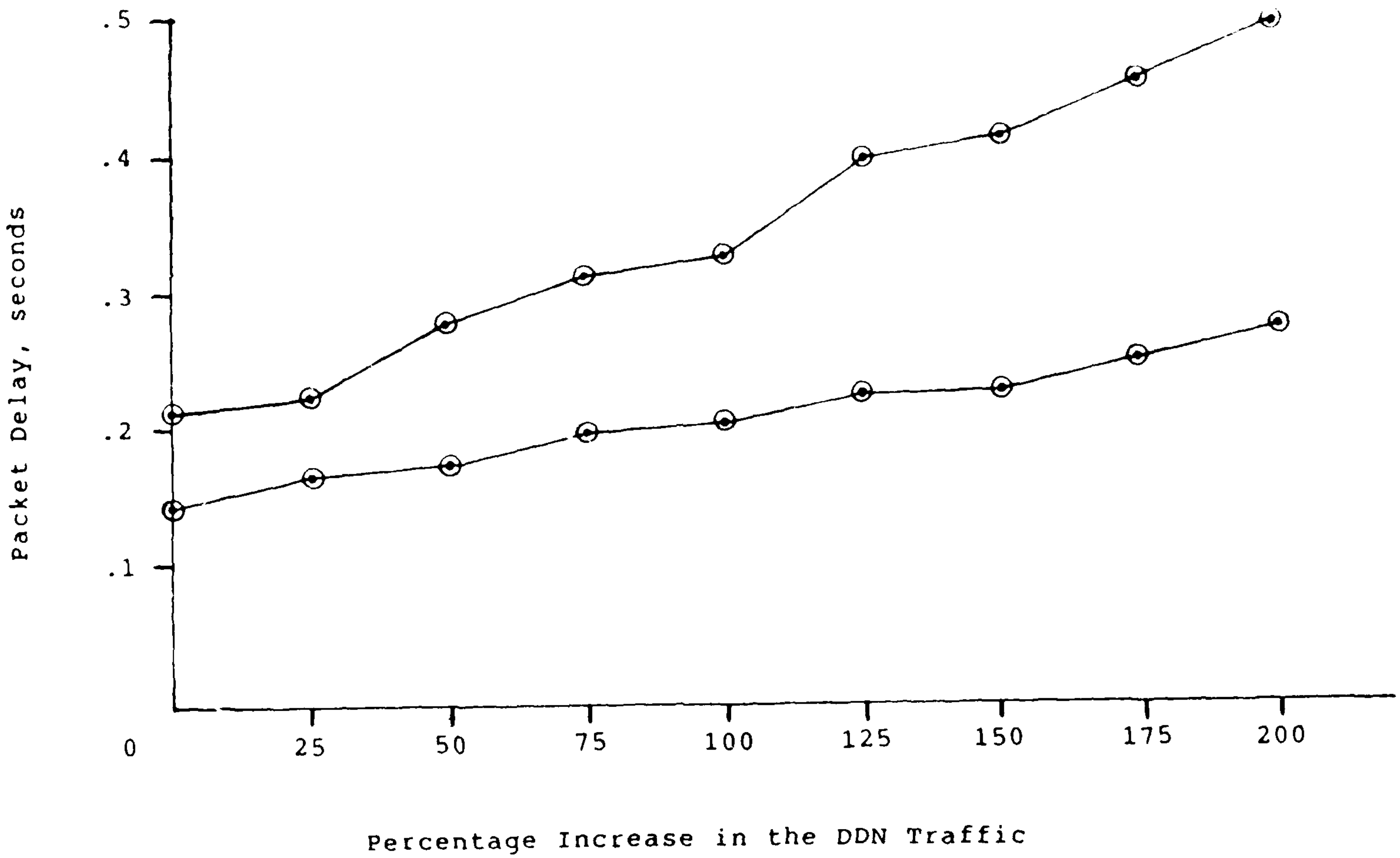


Figure 13. Minimum and Maximum DDN Packet Delay - T1 Links

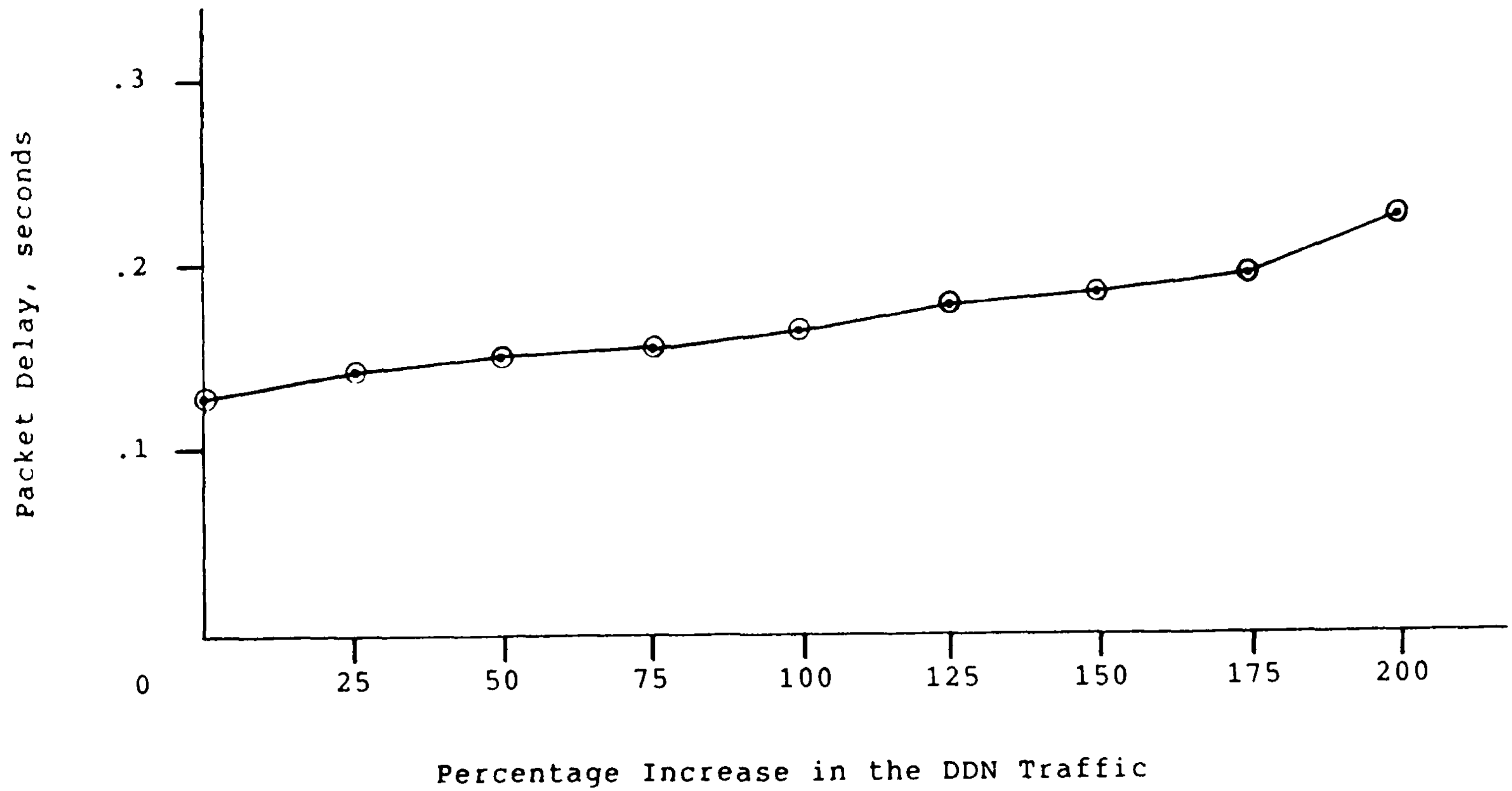


Figure 14. Packet Delay for CIP-to-CIP Connection - T1 Links

.2568 seconds. Additionally, a review of Figure 13 shows that the majority of the overall packet delay is a result of the NAP delays, with the NAP delay constituting over 60% of the upper boundary and over 95% of the lower boundary. Finally, Figure 14 shows that the increased DDN traffic has very little effect on the CIP-to-CIP connections.

Once again a 95% confidence interval is used to determine the limit on the maximum delay. The results of this calculation are summarized in Table 12.

Table 12. Upper Bound with 95% Confidence Interval - T1

Percent Increase	Upper Confidence Bound (seconds)
0	.2547
25	.2812
50	.3199
75	.3448
100	.4061
125	.4479
150	.4625
175	.5217
200	.5349

Applying linear regression techniques to the above data points, the straight line fit for this data is

$$y = (.00149)x + .2484 \qquad \text{(seconds)} \qquad (16)$$

Figure 15 shows the straight line fit.

Discussion of Findings

When reviewing the baseline model, the key consideration is whether the current design can support the workload as spelled out in Chapter Two. A comparison of the requirements

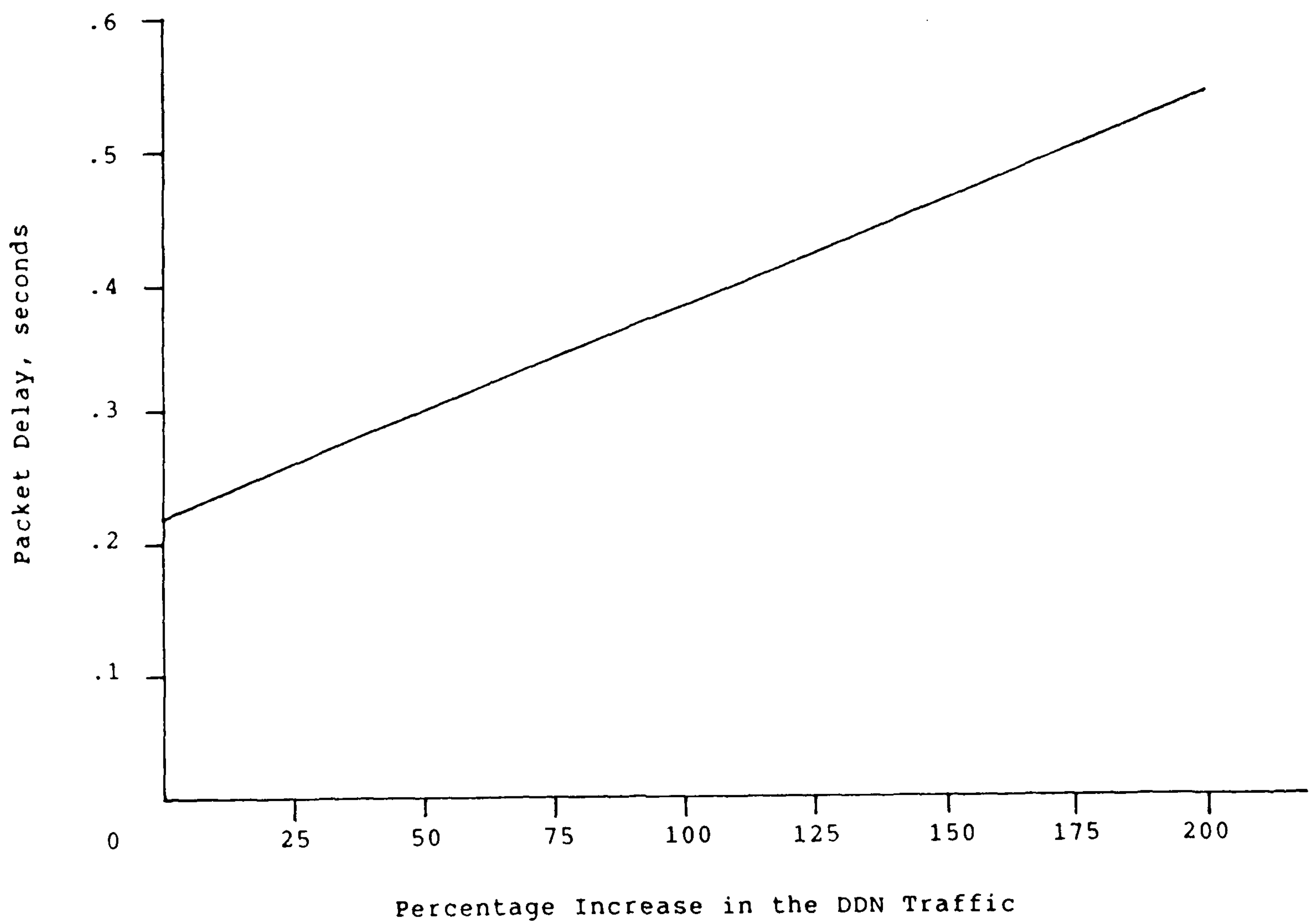


Figure 15. Maximum DDN Packet Delay with 95% Confidence Interval Added - T1 Links

listed in Tables 7 and 8 and those of Table 1 reveals that the Table 1 requirements are much greater than those presented in Chapter Two. Since the original CDS design is based on Table 1, it should be expected that the current design can adequately support the Chapter Two requirements. A look at the baseline model shows this to be true.

First, it is imperative for the interactive users to get a response back from a local system in less than 2 seconds (16). The average delay for a packet was .1342 seconds for a CIP-to-CIP connection and .1732 seconds for a CIP-to-ethernet type of connection. In both cases, the delay is acceptable because even if the input line is a 1200 baud line, the 2 second minimum would be met. For instance, assume a 1200 baud input through a CIP and an output connection via an ethernet. A packet will have a .8533 second delay on the input link, a round-trip delay of .3464 second delay in the CDS, and a .0067 second delay on the first character returning to the user's terminal. Thus, the average user experiences an average 1.2064 second delay between the transmission of a packet and seeing the first character of the response painted on the terminal.

Second, if the CDS is going to act as the primary interface to the ASD computer systems, there must be enough available lines to support the user community. A review of Table 10 shows that even the most heavily used lines are only used on the average less than 60% of the time. Thus, the simulation model verifies that the baseline model will support the Phase I requirements.

The DDN analysis is not as simple as the previous analysis, and before proceeding, an understanding of the current ASD DDN workload is warranted. The DDN traffic on the baseline model only constitutes 50% of the current traffic processed by all the ASD DDN host systems. Thus, if the DDN traffic on the model is increased 100%, the traffic flow is approximately equal to the current ASD DDN traffic.

When trying to look at the DDN statistics, there are several reasons the 2 second rule does not directly apply. First, the model can only account for local delays. Any delay incurred in the DDN is beyond the scope of this effort. Second, according to Dr. Jurick, the 2 second rule only applies to local traffic (16). The average user will tolerate a longer delay when accessing distant systems.

The response of the CDS to increases in DDN traffic is linear over the range of consideration. A comparison of Figures 10 and 11 reveals that the majority of the delay for both the upper and lower boundary is caused by the DDN gateway (NAP). Specifically, the gateway comprises over 80% of the low-end delay and about 50% of the top-end delay. Thus, the NAPs are the key local delaying component for the DDN traffic, but NAP delays also effect local traffic using the internetwork ethernets. The NAPs constitute 75% of the delay for the CIP-to-ethernet local connections. Using the 1200 bps analysis gives a feel for the point at which the increased DDN traffic introduces an unacceptable delay to these types of local sessions. The 1200 baud link alone introduces a .860 second delay. If the CDS causes more than

a .570 second one-way delay, the 2 second design rule is going to be violated. A look at Figure 10 reveals that the DDN packets do not even see a delay this long. Since any local delay is less than this upper boundary, the 2 second rule is not violated for the local traffic.

The delays shown in Figure 10 are approximately one half of the lower boundary delays shown in Figure 9. This result may seem unusual, but there are two reasons for this difference. First, much of the delay incurred by DDN packets is as a result of the DDN gateways. Since the CIP-to-CIP connections do not pass through the NAPs, this major source of delay is avoided. Second, the way the model is developed very few of the DDN logins enter the CDS through the CIPs; therefore, if the local traffic only increases slightly, the delay should act accordingly.

When the DDN link speeds are increased to T1, the overall delay decreases. This makes sense because the increased link processing speed means less time spent in the output queue, but why is the NAP delay increased? Simple, the increased DDN rate also means the DDN gateways are receiving more packets in a given period of time. This also makes sense from the perspective of the model.

The majority of the DDN Telnet sessions created in the model originate locally. This means the logins are to remote systems and since there is a 10:1 ratio between received and transmitted packets, there are many more packets coming from the DDN system than going to the DDN system. Couple this fact with the increased DDN link speed, and it is

understandable that the NAP delay increases as the link speed increases.

Despite the increased NAP delays, the 2 second rule for the local traffic is not violated. Recall from the previous discussion, for a 1200 baud input line the CDS could not introduce more than a .57 second one-way delay. Since the maximum delay is .5349 seconds, the 2 second rule holds.

Finally, the increased link speeds have no affect on the CIP-to-CIP connections. A comparison of Figure 10 and 14, shows that they are basically the same figure. This should be expected since these types of connections do not use the NAPs.

V. Recommendations and Conclusions

Summary

This research effort develops a simulation model of the CDS. The CDS is the primary user interface to the various systems at the ASD Information Systems and Technology Center. The major portion of this effort centers around the development of the different parameters used to drive the model. Since the CDS was not operational at the start of this thesis, statistics from existing systems and projections are used to derive these parameters. The parameters of interest are the number of interactive logins per system, the length of the interactive session, and the amount of data transferred during the session.

Numbers from the CDC and NAS are used to determine the underlying distribution of the interarrival times between logins. A review of these statistics shows this distribution to be exponential. Thus, the login to every system is assumed to have an exponential interarrival time with the mean as determined from existing statistics or the CDS specifications.

The Modcomp statistics are used to prove that the underlying distribution for session length is normally distributed. Statistics from the various systems are used to derive the mean and standard deviation of this normal distribution.

The data transfer distribution is assumed to be normal for all supported systems. The parameters for this assumption are derived from the original CDS specifications.

The number of logins per system, the length of the logins, and amount of data transferred are used to drive the developed CDS model. The model is written using Simulation Language for Alternative Modeling (SLAM) II. The development of the CDS model uses a modular approach with the functions split into six main modules: creating logins, reserving resources, dividing the sessions into transmitted and received packets, routing the packets from node to node using queues to represent the main components of the system, and terminating the packets once they have traversed the network.

The model is used to provide a realistic estimate of the network performance. These estimates answer two questions. Can the current design support the Phase I workload requirements? What effect does increased DDN traffic have on the CDS? Since no login is denied service because of a lack of input or output links, the model shows the CDS can handle the Phase I workload. The model establishes some upper limits on packet delay as the DDN traffic is increased.

Areas for Future Study

The CDS contract includes the delivery of a capacity planning tool. The segments of the CDS simulation model developed during this thesis effort can certainly be used to enhance the capacity planning package. However, the model needs to be expanded in three ways. First, more research on

the input parameters needs to be done once the CDS is operational. Second, the effects on the CDS of loading on the supported systems needs to be investigated. Third, the ability to test the redundant components should be included in the capacity planning tool.

Much of the thesis effort is spent in determining the parameters driving the model and many of these numbers are derived form current systems or estimations by the systems managers. For the capacity planning tool to be effective, these numbers need to be fine tuned. Once the CDS is operational and the user load is representative of the Phase I community, several measurements need to be taken. Some effort should be expended on trying to determine the traffic flow. Basically, there needs to be some verification of the numbers used to drive the current model. But probably the biggest question still needing to be resolved is the per packet service rate of the various CDS components. The numbers used in the CDS model are estimations by the CDS designers and these numbers could drastically affect the output of the model. Therefore, these parameters need to be measured on the real system and used in the capacity planning product.

The second area requiring further study deals with the effects of loading on the CDS. The CDS model treats the supported systems as infinite sources and sinks, but what happens to the packet delay as the supported systems are offered higher and higher workloads. Certainly some delay is experienced at the respective system, but this delay may

ripple back through the CDS. This could have a loading effect on the CDS; therefore, it is important once the CDS is operational to study this effect for possible inclusion in the capacity planning tool.

Both the above suggestions require the ability to take measurements on the CDS or attached systems. This ability should be included as part of the capacity management product. The CDS system manager should be able to chose from a menu the performance parameters to be measured. These actual measurements should in turn feed the capacity management tool.

Lastly, the capacity analysis tool must be able to test the effects of component failures on the operation of the CDS. The CDS design includes some backup components and alternate routing in case of failures. The ability to model these configurations gives the system manager some insight into how much service is deteriorated when the CDS is not fully operational.

Conclusions

The objective of this research is to develop a simulation model of the CDS and this objective is accomplished. The model is an excellent first cut at predicting the performance of the CDS network, and the lessons learned form this effort should be kept in mind when developing the capacity planning tool.

As highlighted on several occasions, any simulation model is only as good as the parameters used to feed the

model. Much of this effort involves collecting and analyzing data to determine these important parameters. While many of the numbers should be refined once the CDS is operational, the underlying distributional proof is valuable information. Any subsequent CDS model can use this effort to argue the interarrival time for logins is exponential and the duration of a session is normally distributed.

The CDS model also gives some valuable insight into the validity of the Phase I design. It shows that the design can support the projected Phase I workload. On the average, an interactive user can expect to get a response within the 2 second delay window. Additionally, there are enough input and output lines to insure the workload is supported without turning away users because of a lack of links.

Finally, the model verifies that SLAM can be used to model complex communications networks. SLAM can use large amounts of memory, but if some care is taken in designing the model, the model can be built within the prescribed memory limitations.

Bibliography

1. Aeronautical Systems Division, Air Force Systems Command. Solicitation F33600-86-R-0322. Wright-Patterson AFB OH, 08 October 1986.

2. Ally A. and S. Bhatia. "Performance Advisor: An Analysis Tool for Computer Communication Systems," Proceedings of the 1986 IEEE Internet Communications Conference. 206-211. IEEE Press, New York, 1986.

3. Architecture Technology Corporation. Area B Network Requirements and Data Flow Analysis. Minneapolis MN, February 1986.

4. Banks, Jerry and John S. Carson II. Discrete-Event System Simulation. Englewood Cliffs NJ: Prentice-Hall Inc., 1984.

5. Barker, Randy, CDS Design Engineer. Personal interview. Control Data Corporation, Dayton OH 30 March through 2 October 1987.

6. CDS Technical Evaluation Team (Jack D. Capehart, Raymond L. Reinhard, and 1st Lt Eric L. Tompkins). Personal interview. Aeronautical Systems Division, Wright-Patterson AFB OH, 30 March through 2 October 1987.

7. Charpantidis, Cosmas P. Specifications of a Simulation Model for a Local Area Network Design in Support of Stock Point Logistics Integrated Communication Environment (SPLICE). MS Thesis. Naval Postgraduate School, Monterey CA, June 1983 (A132142).

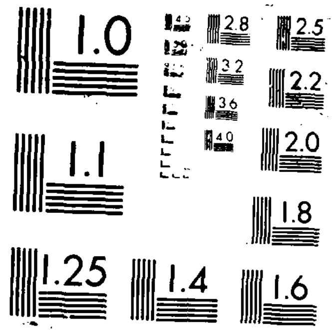
8. Control Data Corporation. Response to Solicitation F33600-86-R-0322 - Part II. Dayton OH, August 1986.

9. Cooper, George R. and Clare D. McGillem. Modern Communications and Spread Spectrum. New York: McGraw-Hill Book Company, 1986.

10. Ferguson, Joseph M., member, CDS Technical Evaluation Team. Personal Correspondence. Aeronautical Systems Division, Wright-Patterson AFB OH, 23 February 1987.

11. Ferrari, Domenico and others. Measurement and Tuning of Computer Systems. Englewood Cliffs NJ: Prentice-Hall Inc., 1983.

AD-A190 682 A SIMULATION MODEL OF THE ASD (AERONAUTICAL SYSTEMS 2/2
DIVISION) CENTRAL DAT..(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. R D SMEDLEY
UNCLASSIFIED DEC 87 AFIT/GE/ENG/87D-61 F/G 12/7 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

12. Foster, Capt Stewart J. Design and Implementation of a Generic Computer Network Simulation System. MS Thesis. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1983 (AD-A127318).

13. Garcia, Albert B. Estimating Computer Communication Network Performance Using Network Simulations. PhD dissertation. University of Dayton, Dayton OH, 1985.

14. Haigh, Peter L. "A Simulation Model for Large Mixed Protocol Networks," Proceedings of the IEEE 1986 International Conference on Communications. Vol 1, 177-181. New York: IEEE Press, 1986.

15. Hughes, H. D. and L. Li. "Simulation Model of an Ethernet," Computer Performance, 4: 210-217 (December 1982).

16. Jurick, Robert, Chief, Scientific Applications Division. Aeronautical Systems Division's Information Systems and Technology Center, Wright-Patterson AFB OH, 30 March through 2 October 1987.

17. O'Reilly, Jean J. "SLAM II, Including a Materia Handling Extension," Proceedings of the 1986 Winter Simulation Conference. 89-94.

18. Papoulis, Athanasios. Probability, Random Variables, and Stochastic Processes. New York: McGraw-Hill Book Company, 1984.

19. Pritsker, A. Alan B. Introduction to Simulation and SLAM II. New York: John Wiley and Sons, 1986.

20. Sauer, Charles H. and Edward A. MacNair. Simulation of Computer Communcation Systems. Englewood Cliffs NJ: Prentice-Hall Inc., 1983.

21. Sauer, Charles H. and others. "Queueing Network Simulations of Computer Communication," IEEE Journal on Selected Areas of Communications, 1: 203-214.

22. Schoemaker, S. Computer Networks and Simulation II. Amsterdam: North-Holland Publishing Company, 1982.

23. Shannon, Robert E. Systems Simulation: The Art and Science. Englewood Cliffs NJ: Prentice-Hall Inc., 1975.

Additional Reading List

1. Hayes, Jeremiah F. Modeling and Analysis of Computer Communications Networks. New York: Plenum Press, 1986.

2. Kleinrock, Leonard. Queueing Systems, Volume 1: Theory. New York: John Wiley and Sons, 1975.

3. McDermott, Capt Steven M. Development of an AFIT ADP System Network Model. MS Thesis, AFIT/GCS/OS/83D-1. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1983.

4. Schwartz, Mischa. Computer-Communications Network Design and Analysis. Englewood Cliffs NJ: Prentice-Hall Inc., 1977.

5. Tanenbaum, Andrew S. Computer Networks. Englewood Cliffs NJ: Prentice-Hall Inc., 1981.

VITA

Captain Richard D. Smedley, Jr. was born on 18 April 1951 in Columbia, Pennsylvania. He graduated from Donegal High School in Mount Joy, Pennsylvania, in 1969 and attended Albright College in Reading, Pennsylvania. In May 1973 he received the degree of Bachelor of Science in Biology. In April 1977 he enlisted in the United States Air Force. After serving two years as a research assistant, he attended Officer Training School. He graduated from OTS in February 1979 as a distinguished graduate. He then attended the Communications-Electronics Officer Course from which he graduated as a distinguished graduate in October 1979. Upon graduation he was assigned to the First Aerospace Communications Group, Offutt AFB as a communications officer. He served in this position until August 1982 when he was selected for the Undergraduate Conversion Program. He attended the University of New Mexico from which he received the Bachelor of Science in Electrical Engineering in May 1984. He was then assigned to the Aeronautical Systems Division's Information Systems and Technology Center, Wright-Patterson AFB, Ohio, as Chief, Advanced Systems Development Branch until entering the School of Engineering, Air Force Institute of Technology, in June 1987.

Permanent address: 313 Red Bud Lane
Wright-Patterson AFB, Ohio 45433

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
AFIT/GE/ENG/87D-61

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
School of Engineering

6b. OFFICE SYMBOL *(If applicable)*
AFIT/ENG

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS *(City, State, and ZIP Code)*
Air Force Institute of Technology (AU)
Wright-Patterson AFB OH 45433-6583

7b. ADDRESS *(City, State, and ZIP Code)*

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
ASD Information Systems & Technology Center

8b. OFFICE SYMBOL *(If applicable)*
ASD/SCEA

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*
ASD Information Systems & Technology Center
Wright-Patterson AFB OH 45433-6503

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
A SIMULATION MODEL OF THE ASD CENTRAL DATACOMM SYSTEM (CDS)

12. PERSONAL AUTHOR(S)
Richard D. Smedley, Jr., B.S., Capt, USAF

13a. TYPE OF REPORT
MS Thesis

13b. TIME COVERED
FROM ________ TO ______

14. DATE OF REPORT *(Year, Month, Day)*
1987 December

15. PAGE COUNT
97

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
12	07	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Computer Communications, Communications Networks, Digital Simulation, Computerized Simulation

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Thesis Chairman: Albert B. Garcia, Lt Col, USA
Assistant Professor of Electrical Engineering

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Albert B. Garcia, Lt Col

22b. TELEPHONE *(Include Area Code)*
(513) 255-3576

22c. OFFICE SYMBOL
AFIT/ENG

DD Form 1473, JUN 86 *Previous editions are obsolete.*

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. ABSTRACT

The purpose of this effort was to develop a simulation model of the CDS, a complex data communications network, using Simulation Language for Alternative Modeling (SLAM) II. The study had two objectives:
(1) Determine if the current design can support the projected Phase I workload. This determination is made after looking at the local packet delay, the availability of input and output lines, and the utilization of the various components.
(2) Determine the effect of increased DDN traffic on the CDS. The specific effects studied are the packet delay within the DDN gateways, the local packet delay, and the total CDS delay on DDN packets.

Since the CDS was not operational during the model development, there were no CDS statistics available to develop the workload; therefore, the input data driving the model was derived from the workload on the current computer systems. This analysis included the basic steps of collecting the data, forming a histogram, making a distributional assumption, and using the chi-square goodness-of-fit test to accept or reject the assumptions.

The CDS simulation model demonstrated the ability of the CDS to support the Phase I workload. Additionally, the model verified that SLAM II can be used to model complex communications networks.

END
DATE
FILMED
4-88
DTIC